Open-set segmentation of images by means of negative examples

Bevandić, Petra

Sveučilište u Zagrebu

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Petra Bevandić

OPEN-SET SEGMENTATION OF IMAGES BY MEANS OF NEGATIVE EXAMPLES

DOCTORAL THESIS

Zagreb, 2023

Sveučilište u Zagrebu

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Petra Bevandić

OPEN-SET SEGMENTATION OF IMAGES BY MEANS OF NEGATIVE EXAMPLES

DOCTORAL THESIS

Supervisor: Professor Siniša Šegvić, PhD

Zagreb, 2023

University of Zagreb

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Petra Bevandić

SEGMENTACIJA SLIKA NAD OTVORENIM SKUPOM RAZREDA PUTEM NEGATIVNIH PRIMJERA

DOKTORSKI RAD

Supervisor: Prof. dr. sc. Siniša Šegvić

Zagreb, 2023.

The doctoral thesis was written at the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Electronics, Microelectronics, Computer and Intelligent Systems.

Supervisor: Professor Siniša Šegvić, PhD

Doctoral thesis contains: 82 pages

Doctoral thesis number: _____

About the Supervisor

Siniša Šegvić was born in 1971 in Split, Croatia. He completed elementary school and high school in Zadar, Croatia, with one year abroad in Milano, Italy. He received the BS degree in electrical engineering (9 semesters) in 1996 as well as the MS and PhD degrees in 2000 and 2004. He has been employed at UniZg-FER as an assistant professor since 2006.

He was a postdoc researcher at IRISA, Rennes and at TU Graz. He led three research projects of the Croatian Science Foundation (MultiCLOD, MASTIF, ADEPT) as well as several industrial research projects funded by Rimac automobili, RoMB technologies, MicroBlink etc. He has participated in the research center of excellence DataCross, several ERDF projects (SafeTram, MAS, A-UNIT) as well as on one FP7 project (ACROSS).

His research and professional interests include computer vision, visual recognition, scene understanding, and dense prediction with deep convolutional models. He has published several papers at top conferences (CVPR, ECCV, NeurIPS) and scientific journals. He has been a reviewer at top conferences (CVPR, ECCV, ICCV, AAAI, ICLR) as well as in scientific journals in the fields of computer vision, intelligent transportation systems and robotics. He participated in the industrial development as a technical consultant. He advises several PhD students funded by EU projects, national projects and private companies. His research group has achieved notable results while participating at computer vision challenges such as WildDash, Robust vision challenge, Cityscapes and Fishyscapes.

Siniša Šegvić speaks English and Italian very well, and has basic communication skills in French. He is married and has three children. He is a member of IEEE.

O mentoru

Siniša Šegvić rođen je 1971. u Splitu. Osnovnu školu i gimnaziju završio je u Zadru osim osmog razreda osnovne škole, koji je pohađao u Milanu. Diplomirao je elektrotehniku na zagrebačkom ETF-u (1996.), gdje je i magistrirao (2000.) i doktorirao (2004.) te se zaposlio kao docent 2006. godine.

Bio je postdoktorski istraživač na institutu IRISA u Rennesu (2006. – 2007.) te na TU Graz (2007. – 2008.). Vodio je tri istraživačka projekta Hrvatske zaklade za znanost (MultiCLOD, MASTIF, ADEPT) te više industrijskih istraživačkih projekata koje su financirale tvrtke Rimac automobili, RoMB, MicroBlink te Promet i prostor. Sudjelovao je u istraživačkom centru izvrsnosti DataCross, na nekoliko ERDF projekata (SafeTram, MAS, A-UNIT) kao i na jednom projektu iz programa FP7 (ACROSS).

Njegovi istraživački i profesionalni interesi uključuju računalni vid, strojno učenje, razumijevanje scena, i gustu predikciju dubokim konvolucijskim modelima. Objavio je 5 radova na

vrhunskim konferencijama računalnog vida i umjetne inteligencije (3xCVPR, ECCV, NeurIPS) te 10 radova u časopisima koje indeksira SCI. Recenzent je na vrhunskim konferencijama računalnog vida i umjetne inteligencije (CVPR, ECCV, ICCV, AAAI) kao i u znanstvenim časopisima u područjima računalnog vida, inteligentnih transportnih sustava i robotike. Sudjelovao je u industrijskom razvoju kao tehnički konzultant. Mentorira više doktoranada koje financiraju europski projekti, nacionalni projekti i privatne tvrtke. Njegova istraživačka grupa postigla je zapažene rezultate na nekoliko natjecanja u računalnom vidu (WildDash, Robust vision challenge, Cityscapes, Fishyscapes i SegmentMeIfYouCan).

Siniša Šegvić odlično govori engleski i talijanski i ima osnovne komunikacijske vještine na francuskom. Oženjen je i ima troje djece. Član je IEEE-a.

Acknowledgement

I am deeply grateful to my advisor, Siniša Šegvić, whose guidance has been instrumental in navigating the exciting and challenging waters of Deep Learning and Computer Vision. His support and encouragement have been essential to the completion of this thesis.

I would also like to express my heartfelt appreciation to all my colleagues from Professor Šegvić's research group. Their valuable technical insights and engaging discussions have enriched the development of my research. Moreover, their camaraderie and help with day-to-day coursework tasks during critical moments made my PhD journey easier and more enjoyable.

Lastly, I cannot thank my friends and family enough. I would especially like to thank my parents for their continuous support throughout my life. Their belief in me has been a constant source of motivation.

Abstract

Open-set recognition simulates real-world settings by introducing outlier samples into model evaluation. This problem is more complex in the dense prediction setting because input can contain a mix of in-distribution and out-of-distribution segments. Open-set recognition tasks can be approached through simultaneous classification and anomaly detection. This work investigates methods for open-set prediction which are trained by means of negative examples. We assume that we train domain-specific models and can thus sample negatives from large and diverse datasets (e.g. ImageNet). These types of negatives introduce additional noise into training because they may also contain some positive content. This thesis will investigate various facets of training on noisy negative samples: from choosing the best architecture and loss to different methods of combining positive and negative samples during batch formation. The presented experiments validate our contributions on standard open-set recognition benchmarks.

Keywords: semantic segmentation, outlier detection, open-set recognition

Segmentacija slika nad otvorenim skupom razreda putem negativnih primjera

Prepoznavanje nad otvorenim skupom razreda simulira rad modela u stvarnim uvjetima uvođenjem izvandistribucijskih primjera u evaluaciju modela. U gustoj predikciji postoji i dodatna razina složenosti jer ulazni primjeri mogu biti mješavina unutardistribucijskih i izvandistribucijskih segmenata. Problem prepoznavanja nad otvorenim skupom razreda možemo rješavati modelima koji istovremeno provode klasifikaciju i detekciju anomalija. U ovome ćemo radu istražiti metode za gustu predikciju nad otvorenim skupom razreda uključivanjem negativnih primjera u postupak učenja. Pretpostavljamo da su modeli koje učimo specijalizirani zbog čega se kao negativi mogu koristiti primjeri iz velikih, raznolikih skupova slika (npr. ImageNet). Takvi negativi unose šum u treniranje s obzirom da mogu sadržavati i unutardistribucijske uzorke. Disertacija će proučiti postupak uključivanja šumovitih negativa u postupak treniranja - od odabira arhitekture modela i gubitka do kombiniranja pozitivnih i negativnih primjera prilikom učenja. Doprinosi će biti vrednovani na standardnim skupovima za validaciju guste detekcije izvandistribucijskih primjera.

U nastavku donosimo sažetak rada po poglavljima.

Uvod

Semantička segmentacija je složen zadatak prepoznavanja koji se odnosi na klasifikaciju svakog pojedinog piksela u jedan od poznatih semantičkih razreda. Razvoj računarstva posljednjih godina omogućio je značajan skok u kvaliteti dubokih modela za semantičku segmentaciju. Time su ovi modeli napredovali od početnih primjena na jednostavnijim problemima do napredne razine izvedbe nad vrlo kompleksnim skupovima podataka kao što su Vistas ili ADE20K. Ovakav impresivan razvoj ukazuje na mogućnost korištenja dubokih modela u mnogim stvarnim primjenama poput autonomne vožnje i medicinske dijagnostike. Međutim, duboki se modeli danas uglavnom testiraju nad zatvorenim skupom podataka gdje slike za testiranje sadrže isključivo one klase koje su viđene u skupu za treniranje. U stvarnom je pak svijetu moguće da se na ulazu nađe nešto s čime se model nije susreo prilikom treniranja - to može biti slika s nepoznatom klasom objekata, ili slika čija je kvaliteta narušena problemima sa sklopovljem ili lošim uvjetima snimanja poput mraka i magle. Iz ovoga se može zaključiti da velika većina trenutnih ispitnih skupova loše modelira stvarne uvjete rada modela.

Predikcija nad otvorenim skupom razreda adresira navedene nedostatke evaluacijom modela nad slikama koje odstupaju od distribucije skupa za treniranje. Pritom se od modela očekuje da strane primjere identificira kao nepoznate i odbije izvršiti klasifikaciju. Jasnije rečeno, od modela se očekuje da klasificira ulazne primjere u skup semantičkih razreda koji je proširen s

dodatnom "nepoznatom" klasom. Identifikacija izvandistribucijskih uzoraka tijekom korištenja modela može dovesti do otkrivanja rubnih primjera te poboljšavanja trenutno korištenih skupova podataka i njihovih taksonomija.

Gusta predikcija nad otvorenim skupom razreda izazovniji je problem u odnosu na predikciju na razini slike jer sadržaj slike može biti u potpunosti nepoznat, u potpunosti poznat ili kombinacija poznatih i nepoznatih dijelova. Posljedično, ispitni skupovi usredotočeni na detekciju negativnih slika ne omogućavaju mjerenje napretka modela za predikciju na razini piksela. Zbog toga je u zadnjih nekoliko godina stvoren niz ispitnih skupova podataka usmjerenih prema segmentaciji nad otvorenim skupom razreda, posebice u domeni obrade slika iz vožnje. Ispitni skupovi WildDash, Fishyscapes i SegmentMeIfYouCan nude različite, međusobno komplementarne pristupe prikupljanju, označavanju i evaluaciji modela za rad nad otvorenim skupom razreda te ukazuju na potrebu za daljnjim istraživanjem ovog problema.

Duboki modeli za gustu predikciju

Umjetne neuronske mreže su modeli strojnog učenja inspirirani strukturom i funkcijom bioloških neuronskih mreža. Procesni element unutar neuronske mreže prima ulazne signale, obrađuje ih i proizvodi izlazni signal. Postoje različite vrste elemenata koji se koriste unutar neuronske mreže, uključujući potpuno povezane slojeve, konvolucijske slojeve i slojeve sažimanja. Elemente umjetnih neuronskih mreža možemo organizirati u slojeve, gdje izlaz jednog sloja služi kao ulaz za sljedeći sloj. Zahvaljujući toj organizaciji, neuronsku mrežu možemo formalno gledati kao kompoziciju jednostavnijih funkcija:
$\mathbf{F}(\mathbf{x},\Theta) = o(f_L(f_{L-1}(\cdots(f_1(\mathbf{x},\Theta_1)),\cdots),\Theta_{L-1}),\Theta_L)$.

Duboke neuronske mreže karakterizirane su velikim brojem slojeva, što im omogućuje učenje hijerarhijskih reprezentacija ulaznih podataka. U računalnom su vidu posebno popularne konvolucijske neuronske mreže jer su prilagođene topološki organiziranim podacima te omogućavaju invarijantnost na operaciju translacije. Ovo je posebno bitno pri razumijevanju slika iz stvarnog svijeta, u kojima je raspored objekata relativno slobodan. Konvolucijske se mreže primarno sastoje od slojeva konvolucije na koje se nadovezuju nelinearne prijenosne funkcije. Uz konvoluciju, često je potrebno u modele uključiti i slojeve sažimanja za povećanje receptivnog polja modela te slojeve normalizacije za stabilniju optimizaciju. Među najvažije konvolucijske pristupe za klasifikaciju slike ubrajamo arhitekture ResNet i DenseNet, koje su ostvarile uspješne rezultate na nizu ispitnih skupova. Modele za klasifikaciju slike moguće je prilagoditi za problem guste predikcije dodavanjem puta naduzorkovanja koji vraća rezoluciju. Ladder-DenseNet je primjer modela za gusto izvlačenje značajki koji uparuje odabranu konvolucijsku okosnicu s memorijski učinkovitim, ljestvičasto povezanim putom naduzorkovanja.

Gusta predikcija nad otvorenim skupom razreda

Zadatak klasifikacije nad otvorenim skupom razreda možemo definirati kao klasifikaciju nad zatvorenim skupom razreda koja je proširena sa zadatkom detekcije anomalija. Neke metode unutar područja detekcije anomalija temelje se na postojećim klasifikatorima (poput onih koji koriste procjenu nesigurnosti) i stoga bi se mogle smatrati metodama za raspoznavanje nad otvorenim skupom podataka, iako ta veza često nije izričito prepoznata. Stoga su u sklopu ovog rada, osim metoda klasifikacije nad otvorenim skupom razreda, razmatrana i druga usko povezana područja poput detekcije izvandistribucijskih primjera, istovremenog treniranja više zadataka te učenja na više skupova podataka.

Detekcija izvandistribucijskih primjera često se temelji na modeliranju distribucije skupa podataka za treniranje. Jednom naučena distribucija omogućava procjenu vjerojatnosti novih ulaznih podataka, pri čemu bi nepoznati primjeri trebali imati manju vjerojatnost od poznatih. Modeliranje ulazne distribucije obično se radi generativnim modelima. Međutim, ti pristupi nisu jednostavno izvedivi ili uspješni u području obrade slike. Postoje strategije za poboljšanje rada tih modela, primjerice prijenosom znanja iz diskriminativnih modela dijeljenjem značajki prilikom učenja.

Raspoznavanje nad otvorenim skupom podataka izvorno se definiralo se kao minimizacija rizika nad otvorenim prostorom podataka pronalaskom funkcije koja može značajno udaljiti nepoznate od poznatih primjera. Odabir latentnog prostora je proizvoljan te su rani radovi su razmatrali obično značajke koje prethode logitima. Ova definicija pretpostavlja da se unutardistribucijski primjeri dokazano javljaju u ograničenom potprostoru latentnih značajki. S obzirom da je ova definicija vrlo uska, u ovom se radu vodimo novijim radovima koji tu vrstu raspoznavanja definiraju kroz prizmu ostvarivanja primarnog cilja, a to je klasifikacija primjera uz odbacivanje nepoznatih ulaza. Ovo nam omogućava razmatranje šireg spektra metoda za detekciju izvandistribucijskih primjera, osobito onih koje se može učinkovito kombinirati s diskriminativnim modelima. Ovo znači da možemo razmotriti i metode temeljene na procjeni nesigurnosti koje odbaciju primjere koji se klasificiraju uz nisku procjenu pouzdanosti predikcije.

Metode temeljene na nesigurnosti lako se prilagođavaju za problem guste predikcije nad otvorenim skupom podataka. S druge strane, jedna je od karakteristika guste predikcije visoka razina nesigurnosti, osobito na semantičkim granicama, gdje susjedni pikseli mogu imati vrlo slične vizualne značajke, a pripadati različitim klasama.

Jedan od načina za poboljšanje detekcije jest proširivanje skupova za učenje dodatnim primjerima. Učenje na dodatnim primjerima može proširiti znanje modela o vizualnom svijetu izvan primarne domene skupa podataka. Nadalje, dodatnim se primjerima mogu poboljšati procjenu nesigurnosti u diskriminativnim modelima i modeliranje gustoće vjerojatnosti podataka kod generativnih modela. Također, veći skupovi za učenje mogu smanjiti osjetljivost modela na po-

mak domene i poboljšati postupak izvlačenja značajki. Pritom podaci kojima proširujemo skup za učenje ne moraju nužno biti stvarni, već mogu biti i sintetički generirani.

Treniranje modela za gustu predikciju putem negativnih primjera

Prepoznavanje nad otvorenim skupom možemo promatrati kao klasifikaciju u kojoj odbacujemo predikcije koje se ne uklapaju u unaprijed definirane kriterije. Na primjer, izlaz standardnih modela klasifikacije može se tumačiti kao vjerojatnost da ulazni uzorak pripada određenoj klasi. Ako je ta vjerojatnost niska, to može ukazivati na to da model gleda strani uzorak i da bi predviđanje trebalo odbaciti. Ovaj se pristup pokazao učinkovitim u klasifikaciji slika i lako se prilagođava semantičkoj segmentaciji. Međutim, predviđanje je nesigurnosti osobito izazovno na semantičkim granicama gdje susjedni pikseli mogu imati gotovo identične značajke, ali pripadati različitim klasama.

Alternativno, problem prepoznavanja nad otvorenim skupom možemo promatrati kao sposobnost modela da kaže da piksel ne pripada niti jednoj od poznatih klasa. Međutim, to nije moguće sa standardnim klasifikatorima koji koriste funkciju softmax na svom izlasu s obzirom da ta funkcija uvijek ima pobjedničku klasu. Drugim riječima, softmax ne može proizvesti vjerojatnost 0 za sve klase. Da bi se to riješilo, model se može modificirati zamjenom softmaxa nad C klasa s C sigmoida, pri čemu svaka sigmoida predstavlja binarni klasifikator odgovoran za identifikaciju jedne klase. Svaki binarni klasifikator implicitno koristi preostale klase iz skupa podataka za treniranje kao negativne uzorke. Ako nijedan binarni klasifikator ne da predikciju veću od praga, možemo pretpostaviti da je uzorak nepoznat. Međutim, ovaj pristup ima smanjuje kvalitetu primarne segmentacijske zadaće i ne radi dobro kada je treniran samo na unutardistribucijskim slikama.

Kao što je ranije navedeno, prepoznavanje nad otvorenim skupom može se prikazati kao klasifikacija, gdje je taksonomija proširena dodatnom "nepoznatom" klasom. Stoga se model može trenirati za predviđanje K+1 razreda proširene taksonomije. Međutim, ovaj pristup ima ograničenje da zahtijeva uključivanje uzoraka za "nepoznatu" klasu u skup za treniranje, što možda nije uvijek lako dostupno ili jednostavno za prikupljanje.

Konačno, prepoznavanje nad otvorenim skupom možemo promatrati i kao istovremenu klasifikaciju i otkrivanje izvandistribucijskih primjera. Izvandistribucijske primjere možemo otkriti evaluiranjem gustoće izglednosti $p(x)$, što može biti računalno složeno, teško se kombinira s primarnim segmentacijskim zadatkom te ne daje dobre rezultate na složenijim skupovima za treniranje kao što su primjerice skupovi s prikupljenim slikama cestovnog prometa. Stoga razmatramo jednostavan diskriminativni pristup gdje detektor izvandistribucijskih primjera oblikujemo kao binarnu klasifikacijsku glavu koji radi paralelno s primarnim glavom klasifikatora s više razreda. Oba klasifikatora mogu dijeliti guste značajke što olakšava njihovu integraciju. Kao i prethodni pristup koji koristi proširenu taksonomiju, diskriminativni pristup detekciji

anomalija pretpostavlja postojanje negativnih uzoraka u skupu za treniranje.

Pokazuje se da uključivanje negativnih primjera u postupak učenja značajno poboljšava rad svih prethodno razmatranih pristupa prepoznavanju nad otvorenim skupom razreda. Negativni se uzorci mogu dobiti uzorkovanjem nekog raznovrsnog pomoćnog skupa za učenje kao što je ImageNet-1k ili ADE20K. Obično nisu dio originalnih podataka za učenje i većinom su izvan distribucije skupa za treniranje. Uključivanje je negativnih uzoraka bitno jer smanjuje vjerojatnost urušavanja značajki. Nadalje, u kontekstu ansambla binarnih klasifikatora, dodatni negativni uzorci povećavaju i poboljšavaju skup protuprimjera za svaki pojedini klasifikator. Za pristupe koji eksplicitno definiraju odvojenu "nepoznatu" klasu, negativni su uzorci nužni kako bi model mogao naučiti izgled "nepoznate" klase i time ju odvojiti od unutar-distribucijkih uzoraka.

Odabir negativnih uzoraka i određivanje najboljeg načina njihovog uključivanja u postupak učenja ključno je za učinkovito gusto prepoznavanje nad otvorenim skupom podataka. Za specijalizirane domene poput vožnje cestom, opći skupovi podataka poput ImageNet-a ili ADE20K-a mogu se koristiti za uzorkovanje negativnih isječaka. Međutim, tako dobiveni negativni isječci mogu biti šumoviti i ponekad sadrže unutardistribucijske dijelove. Ovaj problem možemo ublažiti odgovarajućim strategijama uzorkovanja i otežavanja gubitaka. Takve tehnike mogu značajno smanjiti napor koji bi inače bio potreban za prikupljanje, filtriranje i održavanje negativnog skupa podataka. Nadalje, učenje semantičke segmentacije obično podrazumijeva ugađanje modela predtreniranih na velikim skupovima podataka. Predtreniranje pruža dubokim modelima kvalitetne diskriminativne značajke koje omogućavaju razlikovanje velikog broja razreda. Nažalost, sam postupak ugađanja i specijalizacije na određenu domenu može dovesti do katastrofalnog zaboravljanja čime se smanjuje sposobnost razlikovanja izvandistribucijskih primjera. Efekt se zaboravljanja može umanjiti proširenjem skupa za učenje, primjerice s podskupom izvornih podataka ili u našem slučaju korištenjem općeg negativnog skupa podataka. Stoga prilikom treniranja prednost dajemo inicijalizaciji značajkama dobivenim predtreniranjem na ImageNetu koje su se pokazale bolje za konačnu sposobnost detekcije anomalija od značajki koje su dobivene nakon ugađanja modela za semantičku segmentaciju.

Dodatno, trebamo uzeti u obzir jedinstvene karakteristike pojave anomalija u segmentaciji slika. Anomalije se mogu manifestirati u obliku negativnih cijelih slika ili malih regija unutar ulazne slike. Kako bismo to riješili, moramo osmisliti postupak treniranja koji će se nositi s oba tipa izvandistribucijskih primjera. U osnovi, model može obraditi samo ono što je vidio tijekom učenja. Stoga u svoj skup za treniranje uključujemo i negativne i miješane slike. Miješane slike stvaramo skaliranjem i lijepljenjem negativnih isječaka na unutardistribucijske slike. Budući da se izvandistibucijske regije mogu razlikovati u veličini i položaju, slučajno odabiremo mjerilo i položaj zalijepljenih uzoraka.

Eksperimenti

Predložena četiri pristupa za istovremenu segmentaciju i gustu detekciju anomalija validiramo na nizu eksperimenata. Ekserimente pretežno provodimo učeći na poznatim skupovima podataka koji sadrže slike iz prometa poput Vistasa i Cityscapesa. S obzirom da ograničavamo svoj pristup na relativno usku domenu, negativne podatke uzorkujemo iz skupova podataka opće namjene poput ImageNet-a i ADE20k.

Validacijske eksperimente provodimo uglavnom na skupovima koje smo sami stvorili kombiniranjem skupova podataka iz različitih domena. Za procjenu detekcije negativnih slika uparujemo validacijski skup slika WildDash1 sa skupom slika interijera LSUN. Za procjenu detekcije izvandistribucijskih regija u mješovitim slikama, evaluacijski skup kreiramo lijepljenjem objekata iz skupa Pascal u skup WildDash1. Kreirani validacijski skupovi omogućili su nam da pokažemo sposobnost naših metoda da detektiraju negativne slike i negativne uzorke u slikama mješovitog sadržaja. Iako naš pristup može unijeti određenu pristranost prema detekciji samog postupka lijepljenja, pokazujemo da možemo uspješno prepoznati i semantičke izvandistribucijske primjere u pravim primjerima slika mješovitog sadržaja te na primjerima koji nisu semantički dio negativnog skupa za učenje. Pokazujemo da povezivanje zadatka detekcije izvandistribucijskih primjera sa semantičkom segmentacijom dovodi do bolje detekcije anomalija. Dodatno smo predložili metode za poboljšanje rada dvoglavog modela na malim objektima kroz poboljšano obogaćivanje podataka za treniranje i postprocesiranje izlaza. Naši modeli pokazuju kompetitivne rezultate na testnim skupovima WildDash i Fishyscapes. Uspješno smo primjenili svoj pristup na skupovima podataka StreetHazard, Vistas-NP i UCSD.

Zaključak

Rezultati koji su prikazani u ovom istraživanju podržavaju naše hipoteze da i) korištenje šumovitih negativnih primjera može značajno poboljšati detekciju izvandistribucijskih primjera i raspoznavanje nad otvorenim skupom razreda, i ii) modeli za raspoznavanje nad otvorenim skupom razreda nemaju značajno lošiju klasifikacijsku performansu od modela za raspoznavanje nad zatvorenim skupom razreda. Doprinosi ovoga rada uključuju metodu za učenje modela za detekciju anomalija uz korištenje dodatnih šumovitih negativnih primjera, poboljšanje točnosti modela za gustu detekciju anomalija učenjem na slikama s mješovitim sadržajem te poboljšanje točnosti guste detekcije anomalija dijeljenjem značajki sa standardnim modelom za gusto predviđanje nad zatvorenim skupom razreda.

Segmentacija nad otvorenim skupom razreda je područje s nizom još nerazriješenih izazova, posebno u otkrivanju malih anomalija ili vrlo velikih anomalija. U budućnosti bi bilo vrijedno istražiti nove metode za rješavanje ovih izazova. Jedan od smjerova istraživanja svakako bi trebalo biti iskorištavanje nedavnog napretka u arhitekturama računalnog vida, poput transformera

i kombiniranih modela za vid i jezik. Taj bi se napredak mogao iskoristiti za poboljšanu detekciju izvandistribucijskih primjera i generalizaciju.

Keywords: semantička segmentacija detekcija izvandistribucijskih primjera, prepoznavanje nad otvorenim skupom podataka

Contents

Chapter 1

Introduction

Semantic segmentation is a highly challenging task that involves the classification of each pixel in an image into one of several predetermined classes. The difficulty of this task primarily arises from its computational complexity since we wish to produce high-resolution output and thus need both fine-grained features as well as high-level features that provide significant context. Consequently, early semantic segmentation benchmarks such as Camvid [1] reduced the complexity by keeping the number of classes and image resolution low, and having a relatively uniform image acquisition context. Still, improvements in computer hardware allowed for better methodologies, and larger models with more capacity. Consequently, as simpler benchmarks were being solved, more complex ones replaced them. Today, our models achieve remarkable results even on complex and varied datasets such as Vistas [2] and ADE20k [3].

The impressive progress in semantic segmentation has opened up possibilities for real-world applications, such as autonomous driving [4], road-safety inspection [5], photo-editing [6], and medical diagnostics [7]. However, a crucial limitation of most current benchmarks is that they evaluate in the closed-set scenario. This means that the models are only evaluated for their performance on known classes, while their behaviour in anomalous image regions is not considered. This is a significant drawback for real-world deployment.

Consider for example a very narrowly defined domain such as road driving and one of its most challenging benchmarks - Vistas. This dataset contains 20000 images and 65 classes taken in diverse weather and lighting conditions from various locations. Despite its size, Vistas fails to properly cover all real-life scenarios since it does not include known classes in non-standard poses such as people kneeling or laying, crashed vehicles or fallen vegetation. It also neglects to account for hardware malfunctions, lens distortions, dirt, raindrops or other visual degradations that may affect image quality and model output. Finally, the closed-set evaluation in Vistas does not answer other potentially important questions, such as: What would happen if a model with exemplary Vistas performance encounters anomalous samples that do not conform to the generative process of the training data, such as an elephant or a kangaroo? Likely, the model

would classify them into one of the known classes, potentially leading to critical failures such as inappropriate driving manoeuvres.

One way to fix the described drawbacks of the Vistas benchmark is by creating even more complex datasets. However, this approach is unlikely to bring any solutions in mid-term future as datasets are inherently limited in their ability to capture every possible scenario, even within a specific domain such as road driving. Adding more classes, for example including a wider variety of animals, may seem like a solution, but this can lead to new problems such as long tail distribution, where some classes may be represented with only a small number of samples for these newly added, and likely rarely occurring, classes.

A better way to address the challenge of anomalous input is to adopt an approach that involves rejecting a foreign sample instead of attempting to classify it. From another perspective, we can consider this approach as either classifying the sample either into one of the known classes or into the "unknown" class. This approach essentially expands our taxonomy with a new class that represents the remainder of the visual world. By doing so, the model gains the ability to explicitly indicate that it is unsure about what it is seeing. This inference framework is known as open-set classification. We note that the first formal definition of open-set classification [8] requires minimizing open-space risk by finding a function that guarantees a low prediction confidence in samples distant from the training distribution. In essence, this means that inliers should be contained in a limited region of a chosen feature space. This requirement of inlier representation compactness is fairly strict. In practice, the pursuit of this goal is vulnerable to feature collapse [9, 10] and hence provides no guarantee that the outliers will be distant from inliers in the latent space. On the other hand, Huang et al. [11] offered a simpler definition which emphasized the purpose of open-set classifiers: to correctly classify samples from known classes and accurately identify those from unknown classes. In our work, we adopt the latter view of the open-set recognition problem.

Open-set classifiers are crucial for systems where safety is a critical concern. They allow us to design our systems to respond appropriately to unexpected and potentially dangerous situations. For example, in a road-driving scenario, if the model encounters an exotic animal or if the lens gets dirty, it can indicate the presence of an anomaly and return driving controls back to the driver or initiate the emergency braking procedure. Occurrences of anomalies in real-world settings can be used to discover corner cases, rethink current taxonomies, and further improve datasets.

The need to provide a more realistic representation of the real world led to multiple efforts to design benchmarks aimed at evaluating open-set segmentation performance, particularly in the road-driving domain. The WildDash benchmark [12] includes negative images that are either completely foreign to the road-driving domain or significantly differ from the inlier images, but it fails to account for situations where only a small region of the image is unknown while most

of it is within the inlier distribution. On the other hand, Fishyscapes [13] and SegmentMeIfYouCan [14] benchmarks focus on anomaly detection in mixed-content images. While Fishyscapes tests on artificially created mixed-content images, SMIYC collects real mixed-content images in varied environments. Although artificially created images offer the advantage of virtually unlimited type and position of negative patches, they might be easier to detect due to a narrow inlier domain and pasting artefacts in foreign regions. Currently, the WildDash and Fishyscapes benchmarks evaluate semantic segmentation performance in inlier images but do not measure the impact of outlier detection on segmentation performance in mixed-content images. While these benchmarks have limitations, their development highlights the need for research in open-set classification. The development of both open-set recognition methods and open-set recognition benchmarks should have a mutually reinforcing effect, as has occurred in the closed-set setting. With the creation of more comprehensive benchmarks, researchers will be able to better evaluate the performance of open-set classification models on mixed-content images, leading to further improvements in the field.

This work explores dense open-set recognition, with a particular emphasis on road-driving scenarios. Given the potential practical applications of such models, it is essential that any additional capability for recognizing foreign input comes with minimal overhead in terms of memory and processing time. With this in mind, we present several designs for an open-set recognition module that can be seamlessly integrated with existing dense feature extractors. Our proposed approaches are based on different perspectives of the open-set recognition task.

We can view open-set recognition as classification where we discard outputs that do not fit predefined criteria. For example, the output of standard classification models may be interpreted as the probability that the input sample belongs to a particular class. If this probability is low, that may indicate that the model is looking at a foreign sample and that the prediction should be discarded. This approach has been shown to work well in image classification and is straightforward to adapt to semantic segmentation [15]. However, it faces challenges in the dense prediction context, where there is a lot of inherent uncertainty, especially at semantic borders where neighbouring pixels may have almost identical features but belong to different classes [16].

Alternatively, we can view the problem of open-set recognition as the ability of the model to say that a pixel does not belong to any of the known classes. This, however, is not possible with standard classifiers that use softmax function on model output as this setup ensures that there will always be a winning class. Consequently, softmax cannot produce a probability of 0 for all classes. To address this, the model can be modified by replacing the softmax over C classes with C sigmoids. Each sigmoid represents a binary classifier responsible for identifying a single class. Each of the C binary classifiers uses the remaining classes from the training taxonomy as negative samples. If none of the binary classifiers claim an input sample, we can assume that

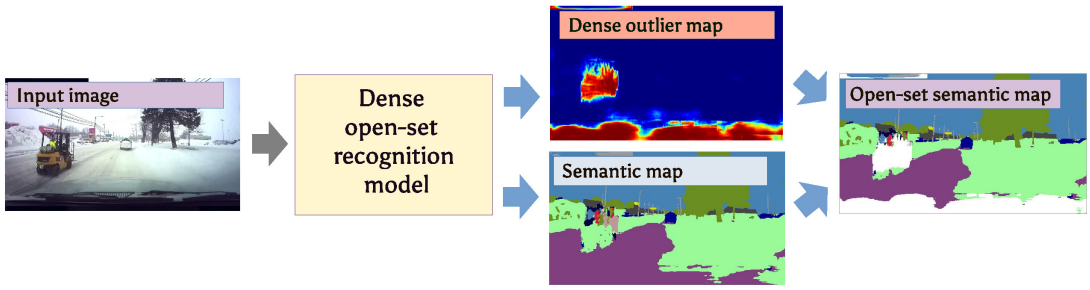

Figure 1.1: A dense open-set recognition model has to correctly identify unknown samples and accurately classify known samples. Such a model should be able to produce: i) a dense outlier map, and ii) a semantic map classifying pixels into C inlier classes. The final output is the merged open-set semantic map which denotes objects foreign to the training taxonomy as a separate "anomaly" class. The example shows open-set segmentation of an image into a Cityscapes taxonomy, where ego-vehicle and the forklift are identified as outliers.

the sample is foreign. However, this approach has some drawbacks. It reduces the quality of the primary segmentation task and does not perform well when trained only on inlier images.

Open-set segmentation can further be framed as C+1 classification, where the taxonomy has been expanded with an extra "unknown" class. Thus, the model can be trained to predict classes from this augmented taxonomy. However, this approach has a limitation that it requires negative training samples for the "unknown" class, which may not always be readily available or easy to collect. Additionally, the sampling of the C+1-st class needs to be carefully balanced during training since it contains a larger visual diversity than the inlier classes.

Finally, we can view open-set recognition as simultaneous classification and outlier detection. Outlier detection involves identifying samples that are foreign to the training distribution and can be applied to datasets without a primary classification task. We consider a simple discriminative approach where we formulate the outlier detector as a binary classifier that works in parallel to the primary multi-class discriminative head [17]. Both classifiers of the proposed two-head approach share dense features which makes their integration seamless and introduces semantically rich features into outlier detection. The downside of the discriminative outlier detection approach is that it assumes the existence of negative training samples.

Figure 1.1 illustrates a dense open-set recognition model that performs outlier detection and semantic segmentation simultaneously on an input image. The outputs of these tasks are then combined to generate a final, dense open-set semantic map.

Note that there are alternative approaches to outlier detection that may be more attractive since, in principle, they do not require a primary classification task [18] or negative data. Principled outlier detectors attempt to model the training distribution $p(x)$ [13, 19]. This is usually done with generative models, though not all of them [20, 21] are suitable for dense prediction context. More suitable generative models are either very hard to train [22] or can only deliver a

lower bound of the likelihood [23] or are outright incapable to infer density [24]. Furthermore, generative models typically fail to capture image semantics, leading to such errors as grouping white cats with white dogs, rather than other, differently coloured cats [25]. They also sometimes behave counterintuitively by assigning higher likelihoods to outliers than inliers [26, 27]. Finally, generative features are often difficult to combine with the primary classification task and ultimately do not perform that well on more complex training domains such as road-driving.

Other approaches to outlier detection rely on image resynthesis with conditional generative models [28, 29]. Outlier samples should be harder to reconstruct which should result in large errors during reconstruction. The dissimilarity between the input and the reconstructed image may then be used for outlier detection. Similarly to generative approaches, resynthesis requires significant computational resources. They have been combined with a primary segmentation task for open-set segmentation and improved efficiency [30], though this approach struggles on complex cluttered scenes.

Among the proposed open-set recognition approaches, the C+1 classifier and the two-head model require negative training examples. C binary classifiers and classification with rejection do not necessarily need additional negative data, but may still benefit from their inclusion into the training process. We thus focus on incorporating negatives into training.

Choosing negative samples and determining how best to include them in the training procedure is an important consideration for effective open-set recognition. For narrow domains like road-driving, general-purpose datasets such as ImageNet [31] or ADE20K can be used to sample negative data. However, these negatives can be noisy and occasionally contain inlier visual content. This noise can be minimized through proper sampling and weighting strategies. Development of training strategies that account for noise reduces the effort that would otherwise be required for the assembly, filtration, and curation of the negative dataset. Moreover, standard semantic segmentation training procedures typically fine-tune models pre-trained on large-scale datasets. This means that pre-trained models usually come with the ability to produce features for distinguishing between a vast number of classes. Keeping the negative samples throughout the training process prevents potential catastrophic forgetting that may arise due to fine-tuning and serves as regularization to prevent overfitting. Therefore, we train our models end-to-end starting from ImageNet-pretrained initialization instead of of pre-trained semantic segmentation models.

In addition, we need to consider the unique characteristics of outlier occurrence in image segmentation. Outliers may manifest in the form of entire negative images or only small regions within an input image. To address this, we need to design the training procedure to handle both types of outliers. Essentially, the model can only handle what it has seen during training. Therefore, we include both negative and mixed-content images in our training dataset. We create mixed-content images by pasting jittered negative samples onto inlier images. Since

outlier patches may vary in size and position, we randomly select the scale and location of the pasted patches.

In order to thoroughly evaluate our approach, we not only test our models on the WildDash and Fishyscapes benchmarks, but also design several validation datasets to address potential drawbacks. One of the primary concerns is the use of real negative data. As it is impossible to expect datasets to cover all real-world scenarios, it is also not possible to have negative data that will cover all possible outliers. However, we show that our approach can successfully detect even those outliers that do not exist in the negative datasets. Additionally, since we rely on mixed-content images during training, there is a risk of the model detecting pasting artefacts instead of proper outliers. Therefore, we explore the ability of our models to detect real outliers and evaluate their sensitivity to the pasting procedure. By doing so, we provide a more rigorous evaluation of the performance of our method.

This thesis is structured as follows. First, we examine deep learning models for computer vision and evaluate their suitability for semantic segmentation. We analyze the fundamental components of these models and investigate how they are integrated into some of the most popular computer vision architectures. We then explore recent advances in outlier detection and open-set recognition, both in the context of images as a whole and in a dense context. We outline our approach to dense open-set recognition, detailing the design of our open-set recognition modules and explaining how they can be effectively trained with noisy negatives. Our experiments on publicly available benchmarks, combined with our validation datasets, offer valuable insights into the strengths and weaknesses of our approach. We complete the thesis by discussing implications of our work and proposing suitable directions for future research in this field.

Our contributions are as follows. We introduce a multi-head open-set recognition model based on sharing features between outlier detection and semantic segmentation of images. We present a technique for assembling training batches of mixed-content images that promote learning of accurate segmentation of out-of-distribution objects. Finally, we propose an optimization procedure that promotes robustness to semantic noise in negative learning examples.

Chapter 2

Deep learning for dense prediction

Artificial neural networks are computational systems inspired by the structure and function of biological neural networks. A single unit within a neural network receives input signals, processes them, and produces an output signal. There are various types of processing elements used within a neural network, including fully connected layers, convolutions and pooling layers.

Processing units in a neural network are usually organized into layers, where the output of one layer serves as the input to the next layer. Due to this organization, a neural network can be formally be viewed as a composition of simpler functions:
$\mathbf{F}(\mathbf{x},\Theta) = o(f_L(f_{L-1}(\cdots(f_1(\mathbf{x},\Theta_1)),\cdots),\Theta_{L-1}),\Theta_L)$.

Deep neural networks are characterized by a large number of layers, which allows them to learn hierarchical representations of the input data.

Some popular deep models for image classification include ResNet and DenseNet architectures, which have achieved state-of-the-art performance on a range of benchmarking datasets. Modifications to these models, such as the introduction of an upsampling path, has been used to improve performance in dense prediction tasks such as semantic segmentation.

The following sections describe the basic building blocks of neural networks and how they are combined into ResNet and the DenseNet architectures. The final section briefly describes how these architectures are incorporated into the Ladder-DenseNet architecture for dense features extraction.

2.1 Basic processing elements

2.1.1 Fully-connected layer

A fully connected layer is an artificial neuron. It is a function that maps the input vector $\mathbf{x}$ into $\mathbf{f}(\mathbf{x}) = \Phi(\mathbf{w}\cdot\mathbf{x}+\mathbf{b})$, where Φ is a non-linear function called the activation function. One specific example of an artificial neuron is perceptron [32] which uses the Heaviside step function as its

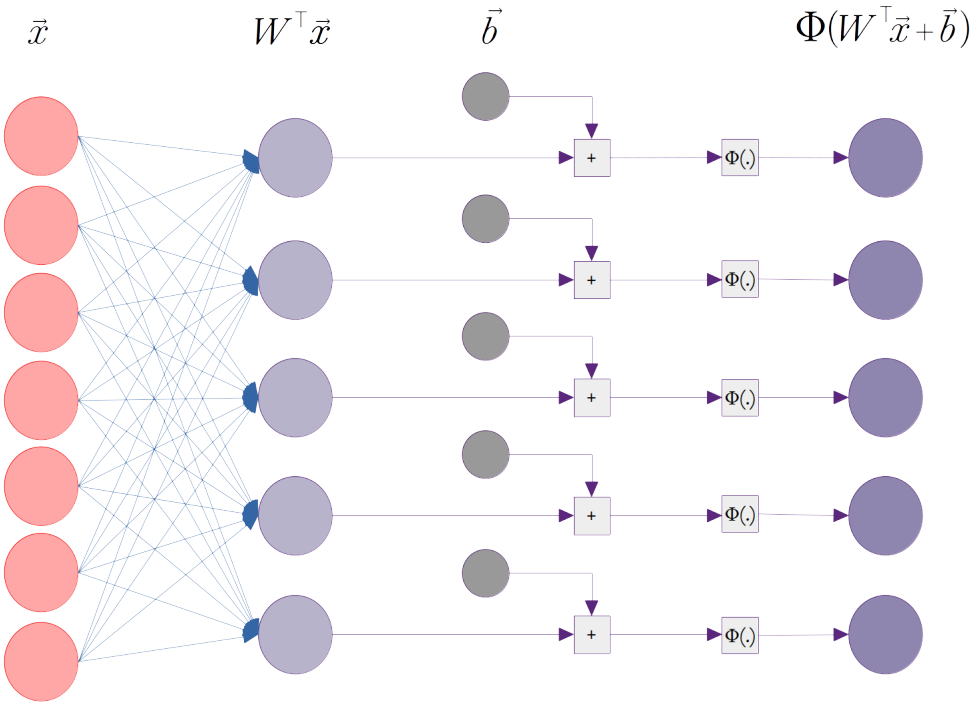

Figure 2.1: Figure shows a fully connected layer. The input vector x is transformed into $\mathbf{f}(\mathbf{x}) = \Phi(\mathbf{w} \cdot \mathbf{x} + \mathbf{b})$, where $\mathbf{w}$ and $\mathbf{b}$ are model parameters, while Φ is a non-linear function called the activation function.

activation function.

Activation functions are necessary since they introduce non-linearity into the model. This is important since image data is complex and not necessarily linearly separable in the input domain. Besides the Heaviside step function, other activation functions include the ReLU function and the sigmoid function. These are preferable to the step function when the model is trained using gradient-based training.

2.1.2 2D convolution

2D convolution is a mathematical operation that combines two 2-dimensional matrices to produce a third 2D matrix that expresses how one of the matrices modifies the other. It can be visualized as a small 2D matrix (known as the kernel or filter) that is "slid" over the image, and the dot product of the kernel and the overlapping portion of the image is computed at each position to produce the output.

The formula for 2D convolution can be expressed as:

$$(K * I)[i, j] = \sum_{m=0}^{k_x} \sum_{n=0}^{k_y} K[m,n] I[i+m, j+n]$$

where I is the input image and K is the convolution filter with spatial dimensions of $k_x \times k_y$.

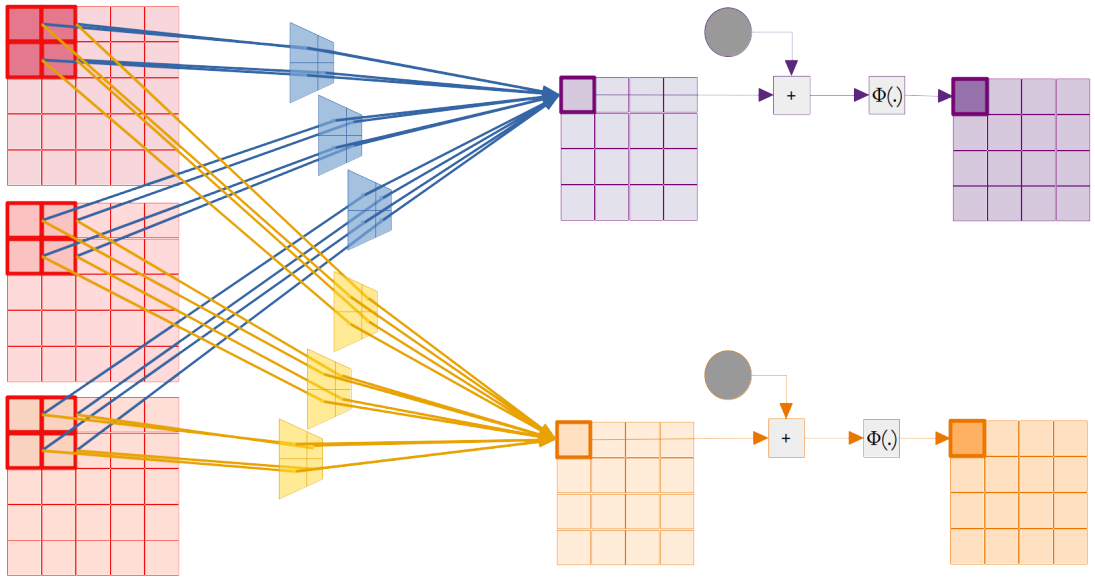

Figure 2.2: Figure shows a convolutional layer. The image is first transformed with a convolution which we visualize as a kernel that slides over an image. The number of kernels is equal to the number of desired output channels. A bias is added for each kernel and the output is passed through a non-linear activation function.

Convolutions are somewhat similar to fully connected layers since both can be expressed as dot products between input and output. However, convolutions have some properties that make them more suitable for image processing. First, they are better at capturing localized features since they operate on small regions of the input image. In contrast, fully connected layers compute a weighted sum of the entire input volume. This also means that convolutions preserve the spatial structure of the data while a fully connected layer does not. Convolutions are equivariant to translation which is especially important in image processing since objects can be located anywhere in an image. Finally, convolutions promote parameter sharing over all locations of the input image which makes them more memory efficient and ultimately more robust to input variations.

2.1.3 Batch normalization

Batch normalization is a technique used in machine learning to improve the performance and stability of neural networks. It works by normalizing the inputs of each layer in a network to have zero mean and unit variance. Specifically, batch normalization involves centering and scaling the activations of each layer using statistics computed over a mini-batch of training examples. This has the effect of reducing the internal covariate shift, which is the change in the distribution of activations in the hidden layers of a neural network due to changes in the parameters of previous layers during training. By reducing this shift, batch normalization can improve the convergence of the network during training and help prevent overfitting.

We define batch normalization as:

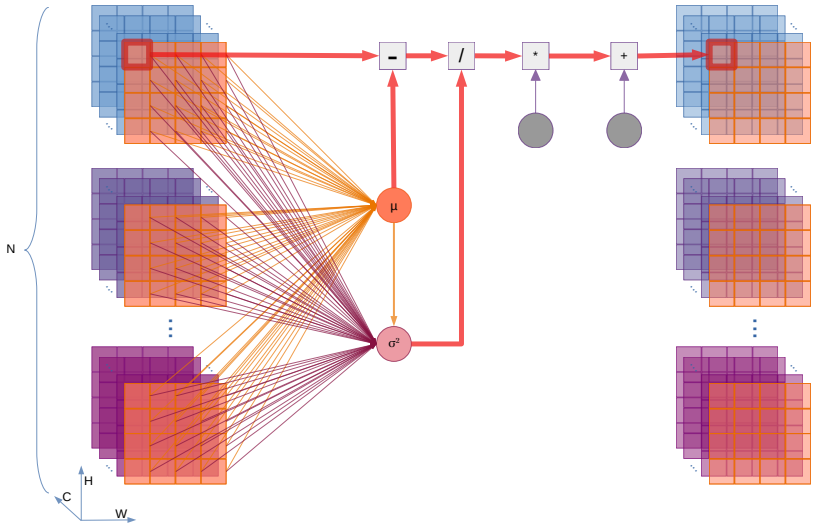

Figure 2.3: Figure illustrates batch normalization. Mean and standard variance are calculated for a channel by reducing over the spatial dimensions as well as the batch samples. The calculated values are used to normalize the input which is then scaled and shifted to produce the final output.

$$\hat{x}^{(k)} = \frac{x^{(k)} - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}, \tag{2.1}$$

where $\hat{x}^{(k)}$ is the normalized output of the k-th neuron in the batch, $x^{(k)}$ is the original input of the k-th neuron, μ_B and σ_B^2 are the mean and the variance of the inputs in the current mini-batch, and ε is a small constant added for numerical stability. The normalized outputs are then scaled and shifted using learnable parameters, which can be denoted as:

$$y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)}, \tag{2.2}$$

where $y^{(k)}$ is the final output of the k-th neuron, and $\gamma^{(k)}$ and $\beta^{(k)}$ are learnable parameters that control the scale and shift of the normalized values, respectively.

During inference, we use population statistics μ_P and σ_P^2 as the dependence on the batch statistics is no longer useful.

2.1.4 Pooling layer

Pooling is a common operation in convolutional neural networks (CNNs) used for down-sampling an image or feature map. Pooling helps to reduce the spatial size of the input, which in turn reduces the number of parameters and the computation required in the network. It also serves to increase the receptive field of neurons in the deeper layers neural networks, which is important for capturing global features and more context.

A special type of pooling is spatial pyramid pooling [33, 34] which divides the input image

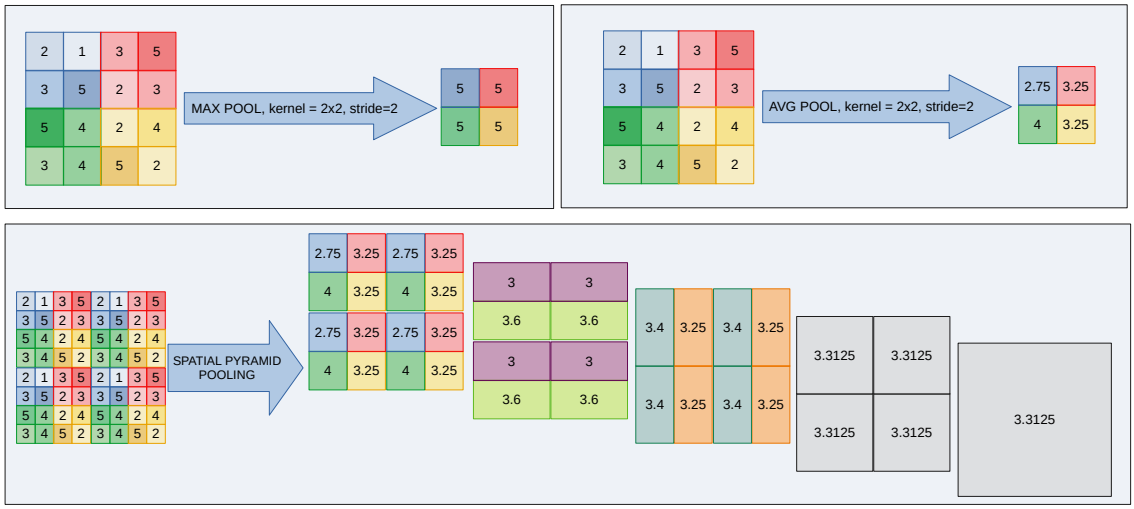

Figure 2.4: We illustrate different types of pooling operations. In standard pooling (top), the output is calculated over a window of a fixed size, e.g. 2x2. The output may either be a maximum value (top left) or the average value (top right) inside this window. Spatial pyramid pooling is a type of layer that performs average pooling over a predefined number of pooling regions. This means that the size of the output is fixed and thus enables the processing of images of arbitrary size.

into a grid of a predetermined number of subregions, or pooling regions, at multiple scales. Then, for each pooling region, the CNN extracts features and applies a pooling operation to produce a fixed-length feature vector that summarizes the content of that region. The feature vectors from all the pooling regions are then concatenated into a single feature vector, which represents the input image. The key advantage of SPP is that it allows a CNN to process input images of arbitrary size, and produce fixed-size feature maps that can be fed into a fully connected layer for classification or other downstream tasks.

2.2 Convolutional neural networks

2.2.1 ResNet

Introduced in 2015, the ResNet [35] uses residual connections to successfully train significantly deeper models than previously possible. This is because residual connections successfully handle the problem of vanishing gradients in very deep networks. Vanishing gradients occur when the gradients become very small as they propagate back through the network during training, making it difficult to update the weights in earlier layers. This can lead to poor performance or even convergence failure.

Residual connections solve this problem by allowing the network to learn residual functions that are easier to optimize. Instead of directly learning the mapping from input to output $y = f(x)$, the network learns a residual mapping, which is the difference between the input and output: $y = f(x) + x$ where x is the input to the layer, $f(x)$ is the mapping function that is

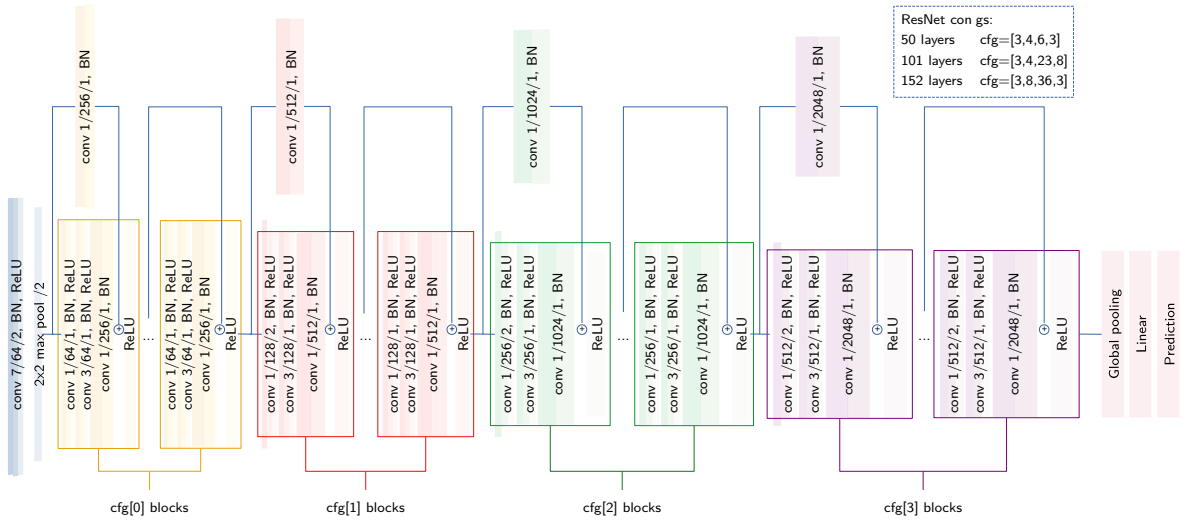

Figure 2.5: We illustrate the ResNet architecture which consists of residual blocks. Each block comprises several convolution and normalization layers. The layer input and output are connected through skip connection and merged with addition. The full network is created through the stacking of residual blocks. The blocks are organized into groups, with spatial resolution halving between two successive groups.

learned by the layer, and y is the output of the layer. We refer to the addition of x to $f(x)$ is the residual connection. There are a couple of potential explanations for the success of residual connections. They introduce paths of varying lengths and behave like ensembles of shallower neural networks thus reducing the problem of exploding and vanishing gradients.

In ResNets, the residual module $f(x)$ consists of three convolutions. First, a 1x1 convolution reduces the dimensionality of the input. This is followed by a 3x3 convolution and another 1x1 convolution to return the original dimensions. There are no pooling layers in Resnet. Instead, certain blocks along the downsampling path perform convolutions with a step of 2.

2.2.2 DenseNet

DenseNet [36] builds on the idea of skip connections introduced in ResNet by taking them one step further. While ResNet only skips one layer at a time, in DenseNet the output of each layer is directly connected to every subsequent layer. This dense connectivity pattern encourages information flow between layers and promotes feature reuse, which helps to mitigate the vanishing gradient problem that can occur in very deep networks.

DenseNet organizes the convolutional layers into blocks, and each block consists of multiple layers. To reduce dimensionality, DenseNet uses 1x1 convolutions before each 3x3 convolution. Additionally, it uses transition layers to reduce the spatial resolution and dimensionality of the feature maps. These transition layers consist of batch normalization, 1x1 convolution, and pooling layers.

In contrast to ResNet, DenseNet uses concatenation instead of addition to combine the out-

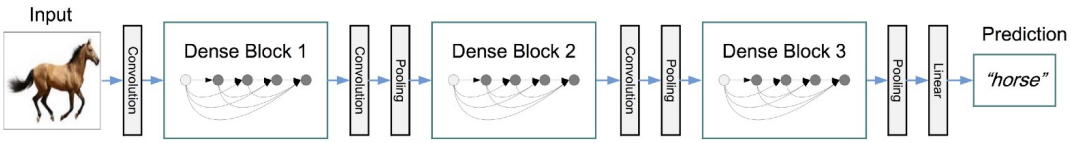

Figure 2.6: DenseNet architecture is made up of several dense blocks. Each dense block contains dense layers that comprise convolution and normalization layers. The input into each dense layer is a concatenation of feature maps of all previous layers. Dense blocks are connected with transition layers which reduce the spatial dimensionality between blocks. The figure is reproduced from the original paper on densely connected models [36].

put of each layer with the input to subsequent layers. This results in a dense connectivity pattern that gives the network its name. The use of concatenation promotes feature reuse and enables the network to learn more discriminative features with fewer parameters, making DenseNet a promising architecture for applications with limited computational resources.

2.3 Semantic segmentation architectures

2.3.1 Ladder-DenseNet

Semantic segmentation requires high-resolution feature maps, which cannot be obtained using standard convolutional neural networks such as ResNet or DenseNet. To address this limitation, the ladder DenseNet [37] proposes an addition of a lightweight upsampling path to a pretrained DenseNet backbone. The upsampling path consists of upsampling blocks that double the spatial resolution of the input using bilinear interpolation and blend it with the projected features of the corresponding dense block output. The proposed upsampling path is lightweight compared to the backbone and memory-efficient, which is crucial for semantic segmentation tasks. To improve memory efficiency even further, the third dense block is split in half, and an additional pooling layer is added in between. This configuration increases the receptive field of the model, allowing it to capture wider context information. Additionally, a spatial pyramid pooling (SPP) layer is included between the backbone and the upsampling path to further capture context information. To improve model generalization, the ladder DenseNet includes auxiliary losses that use soft targets determined as the label distribution in the corresponding N x N window, where N denotes the downsampling factor.

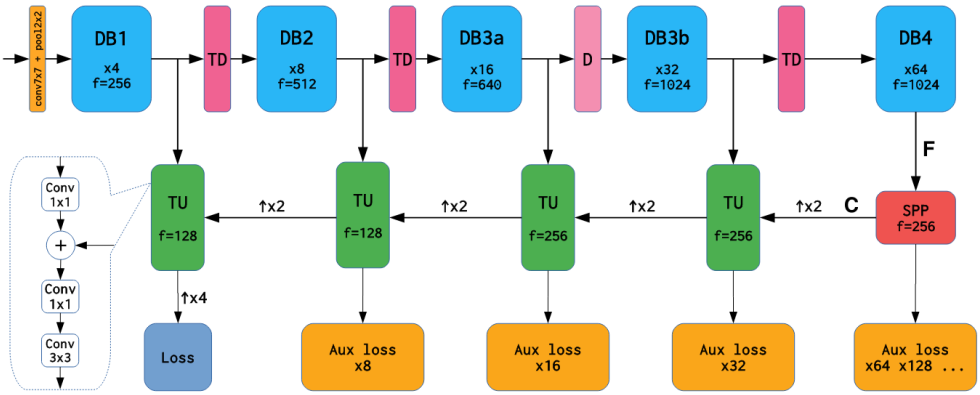

Figure 2.7: Ladder DenseNet architecture uses a pretrained backbone such as DenseNet and replaces the final pooling layer with an SPP block. Spatial resolution is recovered with a lightweight upsampling path which consists of upsampling blocks. Within an upsampling block,input spatial resolution is first doubled and then blended with the corresponding dense block output. Some ladder-densenet variants introduce additional pooling layers in the middle of dense blocks for increased improvement in efficiency. The figure is reproduced from the original paper on ladder densenets [37].

Chapter 3

Dense open-set recognition

The task of open-set recognition extends closed-set classification with anomaly detection. There are methods within the domain of anomaly detection that are based on existing classifiers (such as those utilizing prediction uncertainty estimation) and could thus be considered a part of open-set recognition, though this connection is often not explicitly recognized.

In the following sections, we provide an overview of pertinent literature in the fields of semantic segmentation, anomaly detection, and open-set recognition.

3.1 Outlier detection

Novelty detection is a fundamental problem in machine learning that aims to identify samples that do not conform to the generative process of the training data [38]. This umbrella term encompasses a range of different tasks, including anomaly detection, outlier and out-of-distribution detection, rare-event detection, and one-class classification [25, 39].

A common and principled approach for outlier detection is to model the probability distribution of the training dataset using generative models [19]. This method assumes that anomalous samples would yield low probabilities under the learnt distribution. However, achieving this in practice is challenging, particularly in the field of computer vision [40, 41]. One of the reasons is that existing models can behave counter-intuitively, assigning high likelihoods to outlier samples instead of low ones [26, 42, 43]. Furthermore, generative models tend to have difficulty detecting semantic anomalies without [25]. This might lead to failures such as connecting visually similar objects, such as white cats and white dogs, rather than semantically connected classes such as white and black cats. It is therefore necessary to encourage semantics through alternative methods.

Generative adversarial networks (GANs) [24] are another powerful tool for modelling training distributions and locating anomalies. However, utilizing existing architectures for anomaly detection requires modifications to the training process that ensure proper mapping of images

into latent-space representations [44, 45].

Reconstruction-based anomaly detection is another approach that first encodes and then decodes images [46]. The basic premise of this approach is that a significant difference between the original input and its reconstruction may indicate that the sample is an outlier [47]. An auto-encoder may serve as the generator portion of a GAN architecture, with the discriminator used to evaluate the difference between the input and reconstruction [48, 49, 50]. However, the reconstructed samples tend to be imperfect, regardless of the type of input, making it more effective to use the encoder distribution for anomaly detection [51].

To improve the performance of generative models for novelty detection, it is often necessary to incorporate additional complexity into the training process [52]. For instance, in video sequences, temporal information can be used to improve anomaly detection [53]. Another approach is to leverage alternative tasks, such as future frame forecasting, for better outlier detection [54, 55].

In addition, several studies have demonstrated the benefits of pretraining on a different task or using knowledge distillation [56, 57, 58]. However, it is important to note that the benefits of pre-training can gradually diminish during fine-tuning due to catastrophic forgetting [59]. Therefore, it may be necessary to introduce special loss components to retain the knowledge gathered during pre-training [60].

Taken together, the aforementioned studies suggest that incorporating pretraining and alternative tasks can boost the performance of outlier detection models. Additionally, integrating outlier detection with traditional discriminative models [61] not only improves novelty detection, but also effectively results in open-set recognition models.

3.2 Open set-recognition

The task of open-set recognition was initially introduced by Scheirer et al. [8] with the goal of minimizing open-space risk by learning a latent embedding that can effectively separate unknown samples from known samples by a large distance. The definition of the task is broad and does not specify the space or method for measuring distance. OpenMax [62] was one of the first methods used for training in open-set recognition. It calculates posterior probabilities over C+1 classes based on the distance between the input sample and class representatives in the feature space. Another approach that fits the definition of the task are distance-based classifiers, which classify samples as outliers if they are too far from any training data points in the feature space [63, 64].

The formal definition of open-set recognition can be seen as limiting, as it assumes that the inlier samples are restricted to a particular space. To better align with the practical goal of performing the primary recognition task while also identifying unknown inputs and rejecting

them, it may be more useful to define open-set recognition as a task that combines classification and outlier detection. By doing so, we can expand the range of applicable outlier detection methods. Of particular interest are those approaches that can be integrated with discriminative models, as this will facilitate the creation of open-set recognition models.

A simple way of implementing an open-set classifier is to look at the prediction confidence of a closed-set classifier, assuming, of course, that models produce more confident predictions on known than on unknown inputs. In probabilistic discriminative models we interpret the output as the likelihood of each class given an input sample and and we assign the input sample to the class with the highest likelihood (max-softmax). We can simultaneously treat this likelihood as the model's confidence in its own prediction [15]. However, deep models tend to output highly confident results regardless of the input [65]. Different strategies have been proposed to enhance the informative value of max-softmax. These include recalibration [62, 66], image preprocessing [67, 68], Monte Carlo (MC) dropout [69, 70] and ensembling k 1-class classifiers [58, 71, 72]. Note that there are certain drawbacks to these approaches. While recalibration may improve confidence estimates of the model, it does not necessarily impact the separability of inlier and outlier samples and may not strictly improve outlier detection performance. On the other hand, preprocessing, MC-dropout and ensembling, offer only marginal improvement over the baseline while incurring additional processing cost due to multiple forward passes and models during inference. This may not be acceptable in real-world use cases, especially on large input resolutions and more complex tasks such as semantic segmentation.

There are other, alternative methods for estimating prediction uncertainty. These include examining other properties of the output, such as entropy, or explicitly training networks to recognize difficult examples while only training on inliers. The latter approach can be achieved through joint training of a complementary head in a compound model, allowing the two heads to share the feature representation for increased efficiency and cross-task synergy [16, 61, 73]. However, despite the potential benefits of these methods, they often demonstrate similar behaviour to the max-softmax approach. A principled information-theoretic approach expresses epistemic uncertainty [16] as mutual information between posterior parameter distribution and particular predictions [74]. However, this assumes that MC-dropout is able to approximate Bayesian model sampling, which may not be the case in practice.

Ultimately, the methods that rely on detecting outliers through prediction uncertainty estimation, conflate model uncertainty, which is uncertainty due to the lack of knowledge or "unknown knowns", with distributional uncertainty, which is uncertainty due to the difference between training and testing data or "unknown unknowns". In order to successfully detect outliers, it is important to differentiate between these two types of uncertainties [75].

These approaches can be improved for outlier detection by augmenting the standard discriminative loss with an additional term that encourages high output entropy in negative samples,

such as KL-divergence between predictions and a uniform distribution [76, 77], or a suitable Dirichlet distribution [75]. Another approach is to train a separate prediction head that directly predicts the outlier probability using a negative dataset [17]. However, these methods are sensitive to the choice of the negative dataset, which is challenging to design since prior knowledge of the out-of-distribution space is unknown [78]. To overcome this issue, generative models can be designed to produce synthetic samples at the border of the training distribution [76, 79, 80]. Nevertheless, generating synthetic outliers is only feasible for small resolutions due to the limitations of generative models and GPU hardware. Experiments on small images have shown that diverse negative datasets yield better performance than synthetic outliers [76, 77].

3.3 Training with Negative Data

Focusing solely on inlier samples does not generally yield optimal results in dense open-set recognition. Incorporating negative data during training has been shown to improve outlier detection [60, 72, 77, 81]. Perera et al. propose a template matching approach that uses a shared representation trained simultaneously for ImageNet classification and inlier compactness [60]. However, this method may not be suitable for complex inlier ontologies. Hendrycks et al. propose training max-softmax for low confidence in negative images, which improves outlier detection but does not fully resolve the problem of separating negative samples from those near a semantic border [77]. Chan et al. [81] use COCO images to maximize entropy in negative samples, though they first filter these images to remove inlier classes. Vyas et al. [72] partition the training data into K folds and train an ensemble of K leave-one-fold-out classifiers. The drawback of this approach is that it requires K forward passes.

Including negative data in the training set can have several advantages beyond improved outlier detection. For example, it has been shown to improve the generalization properties of deep models [82]. Alternatively, combining multiple datasets allows for the training of general-purpose closed-set prediction models with extended ontologies [83, 84, 85, 86]. These models are expected to produce better features and be less sensitive to domain shift. Moreover, the extended taxonomies can facilitate outlier detection when these models are applied to narrower subdomains.

Incorporating negative samples into the training set does not necessarily require using real datasets. Previous studies [48, 76] have employed small synthetic negative samples, which have proven effective for detecting small negative images. However, adapting these methods for dense prediction at Cityscapes resolution may not be straightforward [87]. Nevertheless, small synthetic negatives could still be useful for improving outlier prediction in mixed-content images [88].

The soundness of training with negative data has been challenged by Shafaei et al. [89],

who reported underwhelming results for this approach. However, their experiments averaged results over all negative datasets, including the simple MNIST dataset.

3.4 Multi-task Training

Multi-task models involve training a single model to simultaneously perform two or more distinct tasks. They are usually created by attaching several prediction heads on top of shared features [90]. The total loss is usually expressed as a weighted sum of head-specific losses [91] and optimized in an end-to-end fashion.

Multi-task models offer several advantages over single-task models. Combining training datasets creates richer and more varied training data [92]. Evaluation should also be faster since there is no need to run separate models for individual tasks. Feature sharing enables the transfer of knowledge across tasks, which should lead to more robust features and better generalization.

However, when designing multi-task models, it is important to consider task compatibility since not all tasks are necessarily mutually reinforcing [93]. Therefore, it is important to choose tasks that are compatible with each other to maximize the benefits of multi-task learning. There are many task combinations that have been successfully incorporated into multi-task training, including depth, surface normals, and semantic segmentation [94], object detection and semantic segmentation [95], as well as classification, bounding box prediction, and per-class instance-level segmentation [96].

3.5 Dense open-set recognition

Dense open-set recognition remains a relatively under-researched field despite its potentially important applications in a variety of fields such as autonomous driving and industrial facilities. Adapting image-wide outlier detection to the dense context is a non-trivial task due to the added challenge of pixel-level inference. As a result, straightforward adaptation is usually applied to constrained problems. For instance, many existing methods based on generative models [54, 58] are designed for datasets with low image diversity. Similarly, approaches based on image resynthesis have been used to address the more challenging road-driving domain [28, 29, 30, 97], but mostly for specific image segments, such as roads. Furthermore, their computational overhead makes them unsuitable for real-time applications.

On the other hand, adapting open-set recognition approaches to dense prediction is relatively simple and cost-effective [13, 16, 98]. However, as our experiments will also show, most of these adaptations are unable to achieve competitive dense outlier detection performance. We speculate that this is due to the higher aleatoric uncertainty associated with dense prediction. This uncertainty arises from the fact that neighbouring pixels may belong to different classes but

give rise to similar features due to the nature of convolutional models [17]. Consequently, high uncertainty may not necessarily indicate an outlier sample, but rather suggest that the sample is close to a semantic border. Uncertainty estimation may be somewhat improved by moving from pixel-level to component-level estimation [81].

In recent years, the development of proper benchmarks in the field of dense open-set recognition has been a promising advancement. These benchmarks have the potential to drive progress in the field, similar to the way increasingly complex semantic segmentation benchmarks [2, 99, 100] have led to significant developments in the field of closed-set road driving dense prediction. The Wilddash benchmark [12] was a valuable effort that first introduced negative images into dense evaluation but it has some limitations as it does not include mixed-content images. To address this gap, previous iterations of this work [17, 98, 101] crafted a dataset of artificial mixed-content road-driving images with anomalous objects by pasting instances from PASCAL VOC [102]. However, this approach may not be ideal for testing models' outlier detection capabilities since the pasting artefacts can make outlier patches more identifiable. This problem can be addressed in two ways. The Fishyscapes benchmark [13] offers a solution by improving the pasting process with the smoothing of pasted patches. Another approach is to remove one or more classes from existing semantic segmentation datasets and use the removed classes as outliers during testing [17, 88, 103, 104]. However, the drawback of this approach is that the results heavily depend on the choice of removed classes. For example, the BDD-Anomaly dataset uses trains and motorcycles as anomalies, but this may not be the most suitable choice since these removed classes have similarities with some of the kept classes. For instance, trains share many visual features with buses, while motorcycles resemble bicycles. This similarity can cause models to incorrectly identify OOD train pixels as buses, especially if there is not enough contextual information to distinguish between the two. This raises the question of whether this type of mistake should be penalized equally as the error of recognizing those train pixels as a class like 'person.' Similarly, Cityscapes-IDD evaluates on cars (inliers) and rickshaws (outliers) from the Indian driving dataset. However, some parts of some rickshaws can look very similar to bicycles, motorcycles, and cars, which can lead to errors in the detection of anomalies. Vistas-NP [88] attempts to reduce these concerns by removing the entire category of person. These considerations indicate that crafting dense open-set recognition tasks from existing closed datasets is an interesting idea which, however, requires careful planning in practice. Yet another approach is to create a synthetic dataset with anomalous objects [103]. However, real datasets are still considered as better surrogates for real-life operation than synthetic ones. Therefore, the SegmentMeIfYouCan benchmark [14] collected and labeled real-world road-driving images with outlier segments and the Fishyscapes benchmark also includes additional test track and offers dense outlier annotations on a subset of the Lost and Found dataset [105], where the task is to locate very small obstacles on the road.

When evaluating the effectiveness of benchmarks, it is important to consider their evaluation policies. For instance, Cityscapes-IDD suggests training models only on the Cityscapes dataset, which forces models to handle both potential outliers in test images and domain shift between training and test data. On the other hand, other benchmarks do not restrict the training data, making it harder to isolate different factors that affect benchmark performance. Availability of test data is another critical factor to consider. Some of these benchmarks have test images publicly available, while others, like Fishyscapes, keep all test images confidential and require source code submission. Additionally, it is interesting to look at the evaluation metrics used in these benchmarks. Most only measure outlier detection quality using metrics like average precision (AP), area under the receiver operating characteristic curve (AUROC), or false positive rate at a true positive rate of 95% (FPR95). Fishyscapes further measures segmentation quality on the clean Cityscapes validation set but does not evaluate the performance of combined semantic segmentation and outlier detection. It also does not measure semantic segmentation performance on artificial mixed-content images. The Wilddash benchmark measures segmentation quality on negative images, allowing either the outlier prediction or the best inlier prediction. Ultimately, these benchmarks would benefit from incorporating better measures that evaluate proper open-set output [106].

Chapter 4

Open-set training with noisy negatives

We consider efficient models for open-set segmentation that would be suitable for real-time inference on embedded hardware. We intend to learn our open-set models by taking advantage of noisy negative data sampled from ImageNet-1k. Our approach consists of two main components: a dense feature extractor and an open-set recognition module. The feature extractor takes an input image with dimensions H×W×3 and transforms it into a shared abstract representation of dimensions α H×α W×D, where D is the number of output feature channels and α W×α H is the output resolution. The dense open-set recognition module incorporates both recognition and outlier detection, and is trained on labeled inlier images and unlabeled noisy negative images using mixed batches and different losses. We propose that these two tasks be based on shared features in order to promote cross-task synergy and fast inference [60, 73, 92]. Our method is based on two hypotheses: i) training with noisy negatives can improve outlier detection, and ii) discriminative outlier detection and semantic segmentation can share features without significant deterioration of either task [17, 77]. Figure 4.1 illustrates our approach when the Ladder-DenseNet architecture is used as a feature extractor.

The subsequent sections delve into the different components of our open-set recognition approach. Firstly, we provide a detailed description of the design of the feature extractor. Following that, we explain how the training process should incorporate noisy negatives if the goal is to create models that work effectively on mixed-content as well as negative images. Lastly, we examine four open-set module architectures that can be trained efficiently with negative samples.

4.1 Dense feature extraction

Our primary feature extractor is based on the Ladder-DenseNet architecture, which is composed of two main paths: the downsampling path for semantics and the upsampling path for spatial detail restoration. The downsampling path begins with a pre-trained recognition backbone [35,

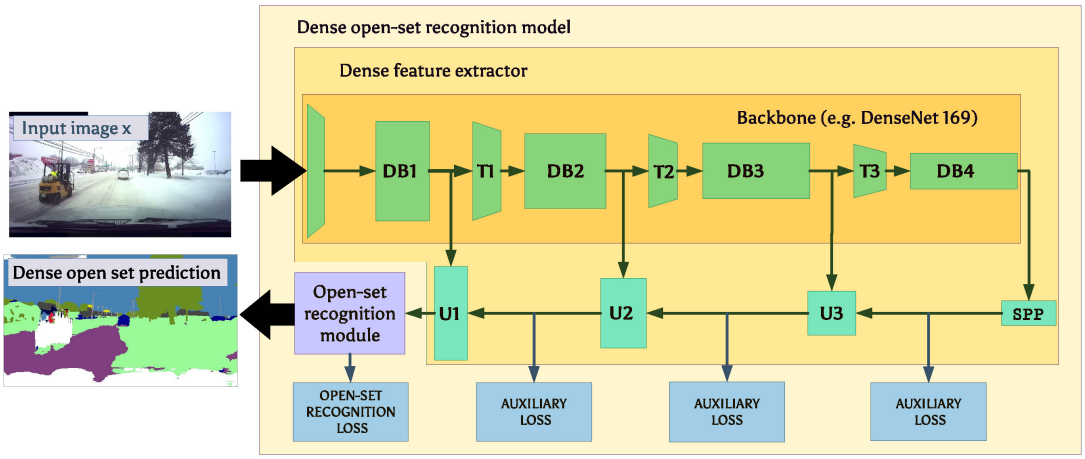

Figure 4.1: Proposed open-set recognition architecture. We use a dense feature extractor (such as ladder-densenet) to acquire dense features. We propagate these features to an open-set recognition module which performs open-set semantic segmentation.

36] and incorporates a lightweight spatial pyramid pooling module [33, 34, 37] for capturing extensive context information. The upsampling path consists of three upsampling modules (U1-U3), which blend low-resolution features from the previous upsampling stage with high-resolution features from the downsampling path. Unlike other encoder-decoder structures [7, 107], the Ladder-DenseNet architecture is asymmetric, with dozens of convolutional layers in the downsampling path and only three convolutional layers in the upsampling path [108, 109]. This makes the architecture memory-efficient and relatively fast while still producing high-quality results. To further reduce memory cost, the features are ouput at 4 times lower resolution.

To improve the segmentation, we use auxiliary losses at 8, 16 and 32 times lower resolutions. The auxiliary loss can be calculated in the following manner:

$$\mathscr{L}_{\text{AUX}} = -\sum_{r\in Ri,\, j\in G_x^r} \sum [\![N_{ij}^r > \tfrac{r^2}{2}]\!] \mathbb{E}_{y_{ij}^r}[\log P(Y_{ij}^r|x)], \tag{4.1}$$

$$y_{ijc}^r = \frac{1}{N_{ij}^r} \sum_{l=ir}^{ir+r-1} \sum_{k=jr}^{jr+r-1} [\![y_{kl} = c \wedge y_{kl} \leq N_{\text{C}}]\!], \tag{4.2}$$

$$N_{ij}^r = \sum_{l=ir}^{ir+r-1} \sum_{k=jr}^{jr+r-1} [\![y_{kl} \leq N_{\text{C}}]\!] \tag{4.3}$$

To obtain the expected output at each resolution, we compute a probability distribution over all segmentation classes. This is achieved by taking the ground truth labels at full resolution and calculating the distribution across the corresponding window. We only consider pixels that have a valid label to avoid including background regions that could skew the results. To incorporate the soft targets at lower resolutions, we use cross-entropy loss between the expected distribution

and the output of the network at each resolution.

4.2 Negative training data

We propose to improve outlier detection by exploiting negative data from an extremely diverse test-agnostic dataset. However, this implies a non-empty intersection between the negative dataset and the inlier training manifold. For example, our negative datasets, ImageNet-1k and ADE20k, contain many classes such as Cityscapes [99] and Vistas [2] (e.g. cab, streetcar, tow truck). Additionally, most stuff classes from Cityscapes (e.g. building, terrain, vegetation) are a regular occurrence in ImageNet-1k backgrounds. Throughout this paper, we refer to this issue as label noise. We believe that label noise can be addressed and managed in any outlier detection approach which trains on negative images.

To mitigate the impact of label noise, we adopt a strategy that involves training on mixed batches with an approximately equal share of inlier and negative images. Since our negative training dataset is much larger than our inlier datasets, we perform many inlier epochs during one negative epoch. This batch formation approach helps prevent occasional inliers in negative images from significantly affecting the training process, and ensures that batchnorm statistics are stable. Our experiments show that this conscientious approach to batch formation successfully promotes resistance to noise in the negative samples.

In our initial experiments, we trained our models on whole inlier and negative images. Although the resulting models performed well on test images with only inliers or outliers, we observed poor performance on images with mixed content. We realized that the outlier detection head needs explicit training on mixed inputs to generalize correctly in such cases. To address this issue, we modified our training procedure by pasting negative images into inlier images during training. More specifically, we resized a negative image to a small percentage of the inlier resolution and randomly pasted it into each training inlier image. This change in training enabled our models to detect outlier objects in inlier contexts. The size of the pasted patches is critical for successful training, and we obtained the best results by randomly choosing the size of the pasted patches from a wide interval, as described in the experimental setup. Our training procedure retains whole negative samples along with mixed content images, enabling the detection of both outlier patches and negative images. Figure 4.2 illustrates this process.

4.3 Open-set recognition module

We propose a dense open-set recognition framework that combines classification and outlier detection using shared features. Our approach assumes access to a labelled inlier dataset $\mathscr{D}_{\text{in}}$ and a noisy negative dataset $\mathscr{D}_{\text{out}}$. Let $\mathbf{x}$ denote an image, $\mathbf{y}$ its corresponding label, $\mathbf{Y}$ the dense

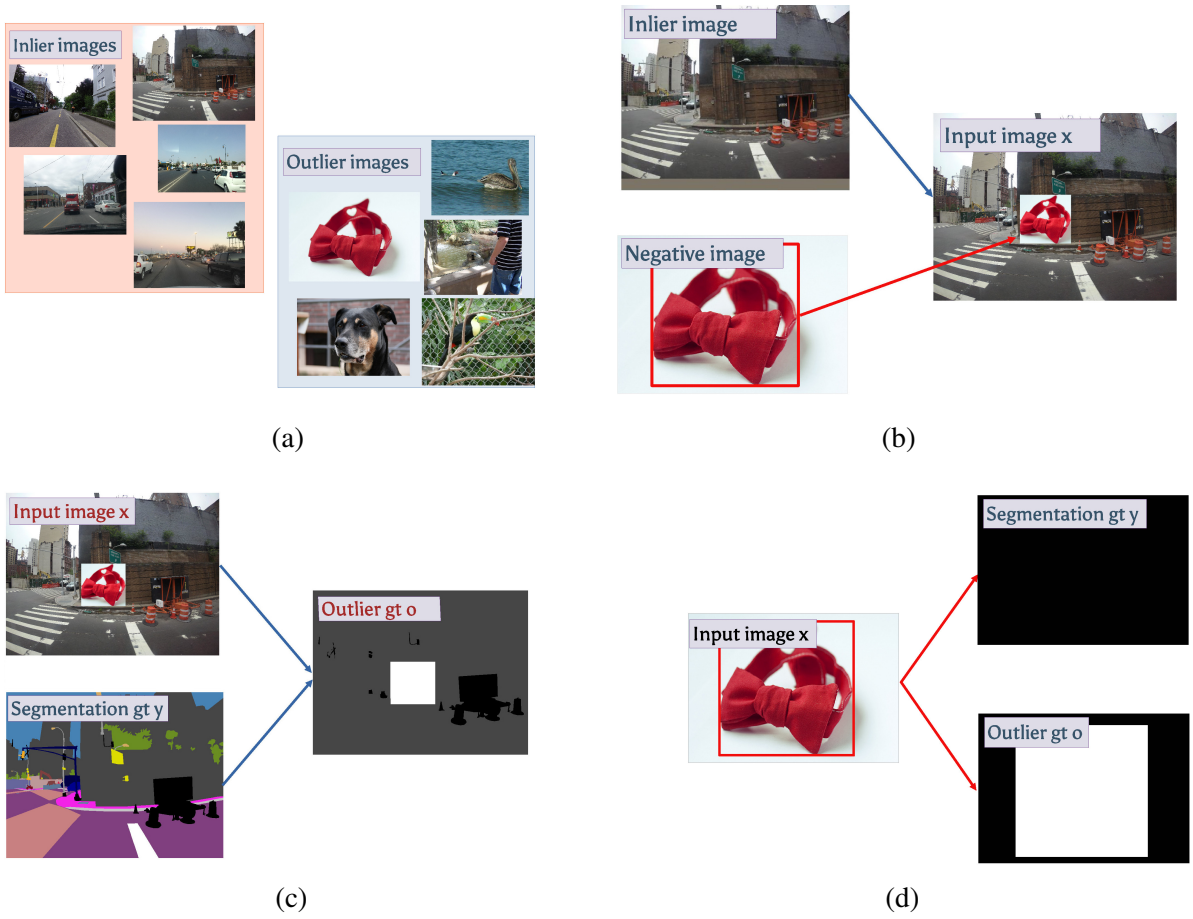

Figure 4.2: We extend our training data with negative examples from a general-purpose dataset such as ImageNet (a). For improved open-set recognition in mixed-content images, we form our training samples by pasting negative samples into inlier images (b, c). We keep some of the negative samples as is, for class balancing and proper out-of-distribution detection in outlier images (d).

predictions over C inlier classes, and $\mathbf{O}$ the dense binary outlier predictions. Our models aim to simultaneously predict the closed-set posterior over classes $P(Y_{ij}|\mathbf{x})$, as well as the probability $P(O_{ij}|\mathbf{x})$ that the sample is an outlier. To achieve this, we use standard cross-entropy losses for both predictions:

$$\begin{aligned}\mathscr{L}_{\text{cls}} &= -\sum_{\mathbf{x},\mathbf{y}\in\mathscr{D}_{\text{in}}}\sum_{ij}\log P(y_{ij}|\mathbf{x})\;;\\ \mathscr{L}_{\text{od}} &= -\sum_{\mathbf{x}\in\mathscr{D}_{\text{in}}}\sum_{ij}\log P(1-O_{ij}|\mathbf{x})\\ &\quad -\sum_{\mathbf{x}\in\mathscr{D}_{\text{out}}}\sum_{ij}\log P(O_{ij}|\mathbf{x})\;.\end{aligned} \tag{4.4}$$

It is worth noting that most of our considerations are applicable to image classification by removing the summation over all pixels (i, j) and considering Y_{ij} and O_{ij} as image-wide predictions.

Under these assumptions, there are four distinct approaches to formulate open-set recognition, as we show next.

4.3.1 C-way multi-class module

The C-way multi-class approach employs a standard classification head, which could be implemented as a 1×1 convolution with C softmax-activated maps. The probability of an outlier is expressed as one minus max-softmax [15]: $P(O_{ij}|\mathbf{x}) = 1 - \max_c P(Y_{ij} = c|\mathbf{x})$. When a negative set is available, this method can be trained to produce a low max-softmax score for outliers [67, 76, 77]. The resulting loss is a combination of the standard inlier loss and the divergence with respect to a uniform distribution on outliers.

$$\mathscr{L}_{C\times MC} = \mathscr{L}_{\text{cls}} + \lambda_{\text{KL}} \cdot \sum_{\mathbf{x}\in\mathscr{D}_{\text{out}}} \sum_{ij} KL\left[\mathscr{U}, P(Y_{ij}|\mathbf{x})\right] \ . \tag{4.5}$$

The drawback of this approach is that there is coupling between recognition and outlier detection. The resulting cross-talk slightly compromises recognition accuracy when the negatives are noisy. Furthermore, this approach does not address the problem of false-positive outliers at semantic borders.

Figure 4.3 illustrates the training of the C-way multi-class open-set recognition module with noisy negatives. Note that both losses need information about the negative samples, the classification loss component so it can ignore them, and the divergence component so it can know which pixels should have the uniform output distribution.

4.3.2 C-way multi-label module

The C-way multi-label approach utilizes C independent heads, each with a sigmoidal activation function. In effect it uses C one-vs-all classifiers [110], where the output of each head is interpreted as the probability of the pixel belonging to the corresponding classes. The final prediction is obtained by selecting the class with the highest probability. The outlier probability is computed as one minus the maximum probability of any class: $P(O_{ij}|\mathbf{x}) = 1 - \max_c P(Y_{ijc} = 1|\mathbf{x})$. The resulting loss function is a sum of the binary cross-entropy losses:

$$\mathscr{L}_{C\times ML} = -\sum_{\mathbf{x},\mathbf{y}\in\mathscr{D}_{\text{in}}} \sum_{ij} \sum_{c} \log P(y_{ijc}|\mathbf{x}) \ . \tag{4.6}$$

The biggest drawback of this approach is the inferior performance on the primary segmentation task.

Figure 4.4 visually depicts the training process for the C-way multi-label module with the

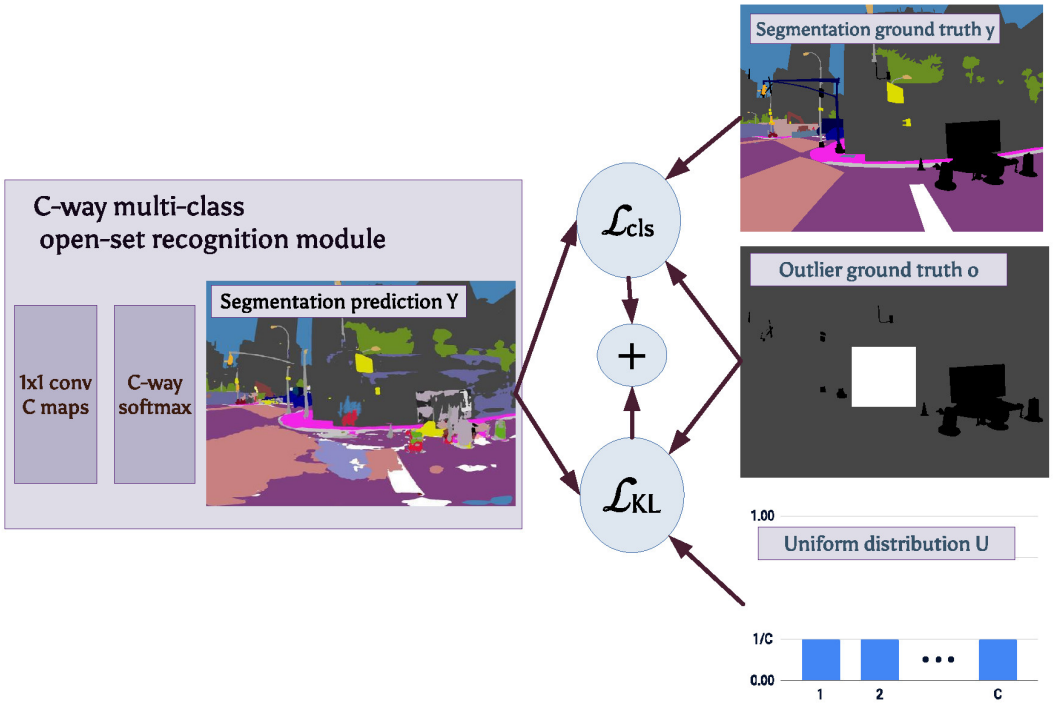

Figure 4.3: C-way multi-class recognition module is trained for semantic segmentation with a standard cross-entropy loss and encouraged to have a uniform distribution in negative samples with KL-divergence.

addition of noisy negative samples. In this approach, each loss component is associated with a distinct binary ground truth map, where pixels belonging to the corresponding class are set to 1. It is worth noting that the remaining classes are used as class 0 in conjunction with the outlier samples. Our experiments show that this increase in type of negative samples seems to lead to better outlier detection performance.

4.3.3 C+1-way multi-class module

The C+1-way multi-class approach incorporates outliers as the C+1th class. While generating the final open-set prediction is relatively straightforward, modelling the outlier probability is not as simple. Merely taking the probability of the C+1st class would ignore its relation to the probabilities of inlier classes. For example, consider a scenario where we have two inlier classes and use the third class as our outlier class. If the output for a given sample is [0.3, 0.3, 0.4], the model would predict that the sample is an outlier. However, if the probability distribution for an input is [0.1, 0.47, 0.43], the model would predict that the sample is an inlier. This illustrates that relying solely on the probability of the C+1st class ignores the contribution of inlier classes to the outlier probability.

It is therefore better to model the outlier probability as a 2-way softmax between the C+1-th logit s_{C+1} and the max-logit over inliers: $P(O_{ij}|\mathbf{x}) = \exp(s_{C+1})/(\exp(s_{C+1}) + \max_{c=1}^{C} \exp(s_c))$.

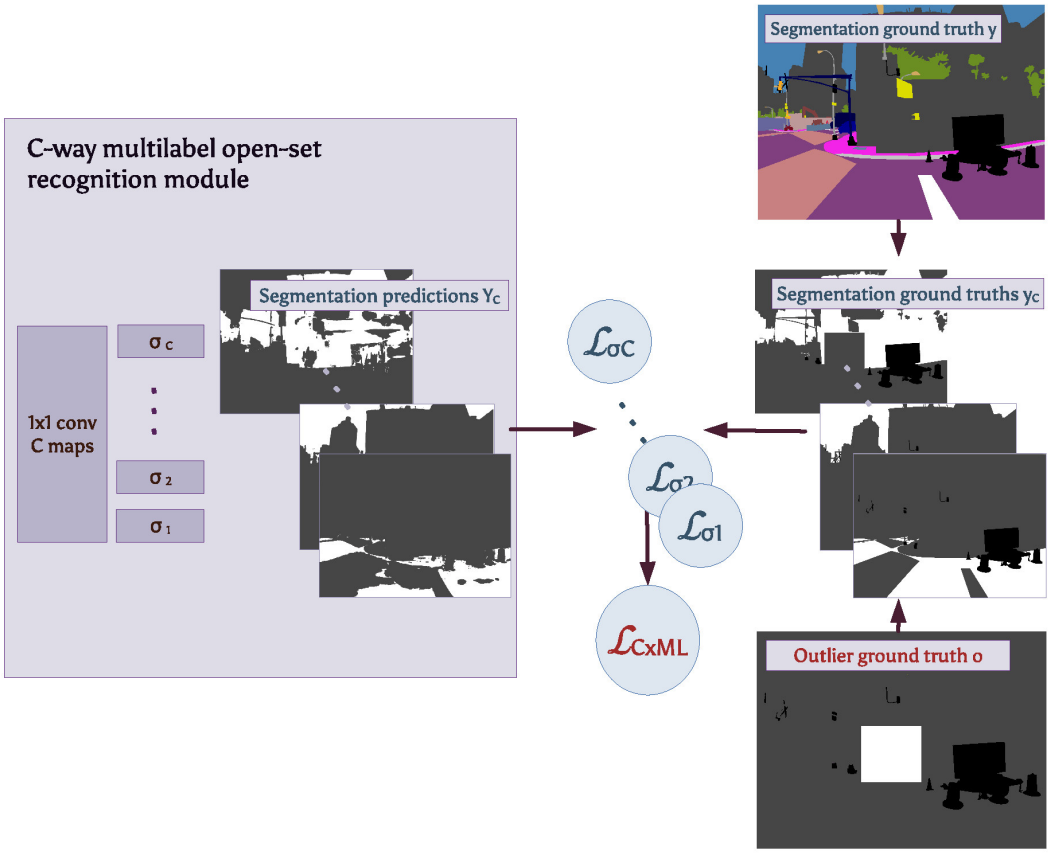

Figure 4.4: C-way multi-label recognition module is constructed as C independent binary classifiers. For each classifier, a ground truth binary mask is created. For each class, both the negative data and the remaining classes serve as negatives.

We account for class disbalance by modulating the loss due to outliers with hyper-parameter λ_{C+1}:

$$\mathscr{L}_{(C+1)\times} = \mathscr{L}_{\text{cls}} + \lambda_{C+1} \cdot \mathscr{L}_{\text{od}} \,. \tag{4.7}$$

Similarly to previous modules, this loss affects inlier recognition, which may be harmful when the negatives are noisy (as in our case).

Figure 4.5 illustrates the training of C+1-way multi-class module with negative data.

4.3.4 Two-head module

Finally, the two-head approach complements the C-way closed-set classification head with a distinct prediction head that is formulated as a binary classifier and directly emits the outlier probability $P(\mathbf{O}|\mathbf{x})$. The classification head is trained solely on inliers, while the outlier detection head is trained on both inliers and outliers. The relative importance of these two heads is

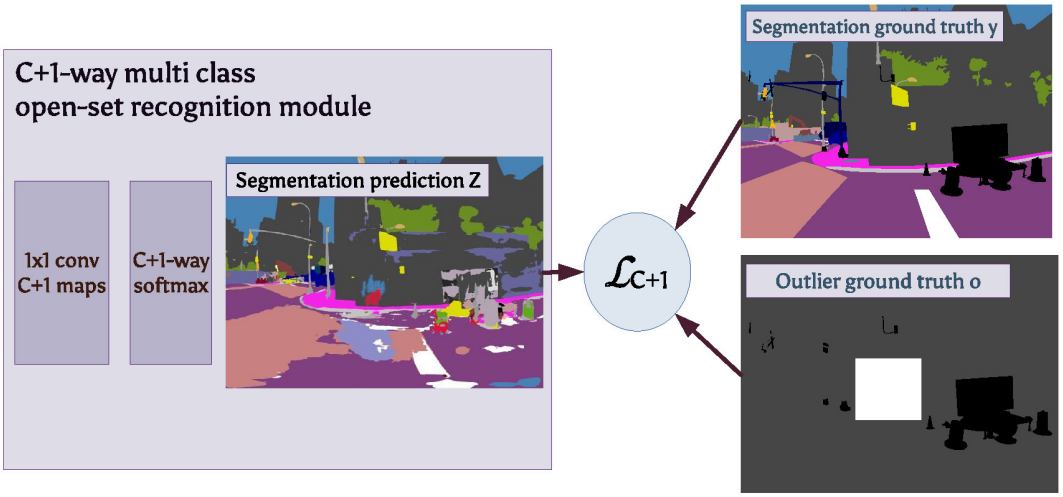

Figure 4.5: C+1-way multi-class recognition module is trained as a standard segmentation module, where negative examples get classified as the C+1-st class.

modulated by λ_{TH}.

$$\mathcal{L}_{\text{TH}} = \mathcal{L}_{\text{cld}} + \lambda_{\text{TH}} \cdot \mathcal{L}_{\text{od}} \,. \tag{4.8}$$

During inference, the outlier detection head takes precedence over the classification head when it predicts that the sample is an outlier. However, it may be possible to modify this inference procedure by introducing a threshold over the outlier probability predicted by the outlier detection head. By setting an appropriate threshold value, the model's sensitivity to outliers can be adjusted, and false positives and false negatives can be controlled. This threshold value can be determined through a validation process or tuned based on the application's specific needs.

Figure 4.6 illustrates the training process of the two-head module. It is essential to note that during training, negative samples are only utilized by the outlier detection head, and the classification head ignores them. This approach ensures that the classification head remains unaffected by noisy and negative data, which preserves the baseline semantic segmentation accuracy even when trained on test-agnostic negatives that are likely to be noisy.

4.4 Loss definitions

Table 4.1 presents our formulation of the different losses needed to implement various open-set detection modules. We use the following notation: x is the input image, y is the ground truth segmentation into N_{C} classes, and o is the ground truth indicating whether a pixel is an

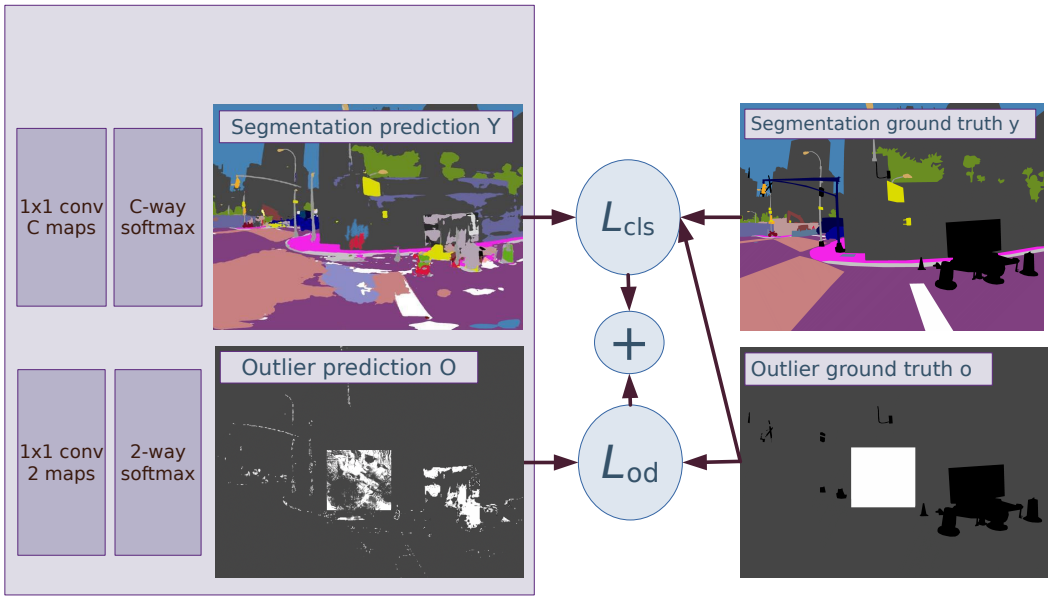

Figure 4.6: The two-head recognition module comprises two heads. The first head performs semantic segmentation while the second performs binary classification into inliers and outliers. The two losses are added to form the final training loss.

inlier (labelled 1) or an outlier (labelled 0). During training, some pixels may be ignored, such as those belonging to the ego-vehicle. To indicate that these pixels should be ignored, they are given a distinct label, for example a number greater than the number of classes N_C in the ground truth segmentation image y or label 2 in o. In the C+1-way multi-class setup, N_C equals C+1, and both inlier and outlier pixels are labelled with 1 in o.

Table 4.1: Losses used during training, with the following assumptions: x is an input image, y contains ground truth segmentation into N_C classes and o contains ground truth indicating whether a pixel is an inlier or an outlier. Dimensions of x, y and o are $H \times W$.

Loss	Expression
multi-class classifier loss	$$\mathscr{L}_{\mathrm{MC}} = -\sum_{i,j \in G_x} \lambda_{y_{ij}} [\![o_{ij} = 1 \wedge y_{ij} \leq N_C]\!] \log P(Y_{ij} = y_{ij} \mid x),$$ $$P(Y_{ij} = y_{ij} \mid x) = \frac{\exp s^{ij}_{y_{ij}}(x)}{\sum_c \exp s^{ij}_c(x)}$$
multi-label classifier loss	$$\mathscr{L}_{\mathrm{ML}} = -\sum_{i,j \in G_x} \sum_{c=1}^{N_c} [\![y_{ij} \leq N_C]\!] \Big([\![y_{ij} \neq c \vee o_{ij} = 0]\!] \log \frac{1}{1 + \exp s^{ij}_{y_{ij}}(x)} + [\![y_{ij} = c \wedge o_{ij} = 1]\!] \log \frac{\exp s^{ij}_{y_{ij}}(x)}{1 + \exp s^{ij}_{y_{ij}}(x)} \Big)$$
auxiliary loss	$$\mathscr{L}_{\mathrm{AUX}} = -\sum_{r \in R} i,j \in G^r_x [\![N^r_{ij} > \tfrac{r^2}{2}]\!] \mathbb{E}_{y^r_{ij}} [\log P(Y^r_{ij} \mid x)],$$ $$y^r_{ijc} = \frac{1}{N^r_{ij}} \sum_{l=ir}^{ir+r-1} \sum_{k=jr}^{jr+r-1} [\![y_{kl} = c \wedge y_{kl} \leq N_C]\!],$$ $$N^r_{ij} = \sum_{l=ir}^{ir+r-1} \sum_{k=jr}^{jr+r-1} [\![y_{kl} \leq N_C]\!]$$
outlier detection head loss	$$\mathscr{L}_{\mathrm{TH}} = -\sum_{i,j \in G_x} [\![o_{ij} \leq 1]\!] \log P(O_{ij} = o_{ij} \mid x)$$
Kullback Leibler divergence	$$\mathscr{L}_{\mathrm{KL}} = \sum_{i,j \in G_x} [\![o_{ij} = 0]\!] \mathrm{KL}(\mathscr{U} \parallel P(Y_{ij} \mid x))$$
confidence loss	$$\mathscr{L}_{\mathrm{C}} = -\sum_{i,j \in G_x} [\![y_{ij} \leq N_C]\!] \log(c_{ij} \mid x)$$

Table 4.2 illustrates how the aforementioned losses are combined to achieve the open-set models.

Table 4.2: Total training losses

Model	Total loss
C× multi-class, C+1× multi-class	$(1-\lambda_{\text{AUX}})\mathscr{L}_{\text{MC}}+\lambda_{\text{AUX}}\mathscr{L}_{\text{AUX}}$
C× multi-label	$(1-\lambda_{\text{AUX}})\lambda_{\text{ML}}\mathscr{L}_{\text{ML}}+\lambda_{\text{AUX}}\mathscr{L}_{\text{AUX}}$
C×multi-class with outliers	$(1-\lambda_{\text{AUX}})(\mathscr{L}_{\text{MC}}+\lambda_{\text{KL}}\mathscr{L}_{\text{KL}})+\lambda_{\text{AUX}}\mathscr{L}_{\text{AUX}}$
two heads	$(1-\lambda_{\text{AUX}})(\mathscr{L}_{\text{MC}}+\lambda_{\text{TH}}\mathscr{L}_{\text{TH}})+\lambda_{\text{AUX}}\mathscr{L}_{\text{AUX}}$

Chapter 5

Experiments

In this chapter, we present the results of our validation and ablation experiments and provide additional insights into our method. We begin by describing the used datasets in greater detail. We then explain the evaluation metrics we used to measure the quality of semantic segmentation and outlier detection. We also provide a precise definition of the losses we used to implement the considered open-set recognition modules. Next, we provide details about the experimental setup, training, and inference procedures. Finally, we present the results of our validation and testing experiments.

5.1 Datasets

5.1.1 Cityscapes

The Cityscapes dataset [99] is a widely used benchmark dataset for evaluating vision algorithms. It comprises images captured from the perspective of a driver in various cities across Germany and neighbouring countries. The dataset offers high-quality instance and semantic level segmentations for its 2-megapixel images, which are split into train, validation, and test sets. While the dataset offers 30 classes at the semantic level, only 19 classes are considered for evaluation. The images were captured during the daytime, in different seasons, and under predominantly good weather conditions. However, this uniformity in acquisition conditions is also a limitation of the dataset, often leading to the overfitting of models. To mitigate this issue, it is common to complement the Cityscapes dataset with other road-driving datasets.

Training and validation subsets consist of 2975 training and 500 finely annotated images, respectively. The dataset also provides 20 000 coarsely annotated images, which were not used in our experiments.

5.1.2 Vistas

The Vistas dataset [2] is another dataset containing images of traffic scenes. Unlike the Cityscapes dataset, Vistas is larger and more diverse, with images from all over the world taken in different weather conditions and from various viewpoints. The dataset provides images with varying resolutions, with an average of 8.5 megapixels, and includes 65 semantic classes. To enable simultaneous training on Vistas and Cityscapes, it is common to first map the labels into a common taxonomy. In our experiments, we remap the labels into the Cityscapes taxonomy, which is possible since all Vistas classes are either subclasses of Cityscapes classes or completely unrelated to them. During training, we ignore the unrelated Vistas classes.

There are 18000 training images and 2000 validation images. In addition, there are 5,000 publicly available unlabeled images, which we did not use in our experiments.

Vistas-NP

One way to create a validation dataset with proper outlier samples is by removing a subset of classes from training. We follow [88] to create Vistas-NP by removing the "person" category. This arrangement excludes an entire category and therefore features a very small overlap between inlier and outlier classes, unlike BDD-Anomaly [103] and Cityscapes-IDD [104].

This results in the training subset of 8833 images collected from Vistas trainval. We use the remaining 11167 images to create Vistas-NP test.

5.1.3 ImageNet

ImageNet [111] is a vast image database manually annotated using the WordNet schema. The database includes images of different sizes, with no specific object location and the majority are in colour. However, they are relatively small, with an average resolution of 0.2 megapixels.

The dataset provided for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [31] contains a subset of images classified into 1000 classes, with provided bounding boxes around objects in about half of the images. It is divided into training, validation and test datasets, with the test dataset being unlabeled. The training set contains over one million images, while the validation set and test set contain 50,000 and 100,000 images, respectively.

In our experiments, we utilize the ILSVRC dataset to construct two sets of negative samples. The first collection, which we will denote with ImageNet-1k, includes all the available images in their entirety. The second collection, denoted as ImageNet-1k-bb, includes only the images with provided bounding boxes and utilizes only the content of the first bounding box for negative samples.

5.1.4 ADE20k

ADE20K [3], also known as the MIT Scene Parsing Benchmark, is a comprehensive dataset for scene parsing tasks. It provides dense annotations for 150 semantic categories in a variety of indoor and outdoor settings. The images in this dataset are relatively small, with an average resolution of around 0.2 megapixels. ADE20K consists of 20,210 training images and 2,000 validation images,

Although the ADE20K dataset is smaller and less diverse than the ILSVRC dataset, it still offers sufficient variability to collect negatives in the road-driving context. Unlike ILSVRC, ADE20K is labelled at a pixel level, which allows us to cut and paste non-rectangular negative patches. Our experiments show that using a variety of pasted shapes increases precision and improves the performance of our open-set models.

5.1.5 WildDash 1

The WildDash 1 [12] dataset provides a benchmark for semantic segmentation and instance segmentation. It complies with the Cityscapes labelling policy. It focuses on providing performance-decreasing images. These images are challenging due to conditions and unusual locations in which they were taken or because they contain various distortions.

The images are divided into a validation set and a test set. There are 70 validation and 156 test images. The test set contains 15 images which are marked as negatives. All pixels in these images are considered out-of-distribution in the context of semantic segmentation on road-driving datasets.

WD-Pascal

To evaluate the performance of our models on mixed-content images, we created a new dataset called WD-Pascal by randomly pasting animal classes from the Pascal dataset [112] into WildDash validation images. We selected animals that occupy at least 1% of the WildDash image area to ensure their visibility. Since WD-Pascal is created at test time, its size is virtually unlimited. To account for the variability in the pasted content, we performed 50 assays across WildDash val with different types, sizes, and positions of the pasted animals. To ensure reproducibility, we fixed the random seed generator for all assays.

WD-LSUN

In order to assess the performance of our models on negative images, we augmented WildDash validation images with random subsets of LSUN images [113]. We chose the number of LSUN images so that the share of inlier and outlier pixels was approximately equal. As the choice of

LSUN images is random, we fixed the random seed generator in our experiments and conducted 50 assays across WildDash validation.

5.1.6 Fishyscapes

Fishyscapes [13] is a dataset designed for dense anomaly detection in a road driving context. The dataset consists of three tracks: Fishyscapes Static, L&F, and Fishyscapes Web. The Fishyscapes Static track contains Cityscapes images with pasted anomalous objects from Pascal VOC. The pasting is done with smoothing applied, and there are 30 validation images and 1000 hidden images available. The L&F track is a subset of Lost and Found images with true anomalies. The images are filtered to have no overlap between Cityscapes classes and anomalies and are relabeled by the authors. This track has 100 validation images and 275 test images. The Fishyscapes Web track includes objects pasted into Cityscapes images that are crawled from the web. New versions are added iteratively over time.

5.1.7 UCSD

The UCSD dataset is a widely used benchmark for anomaly detection in crowded scenes, which consists of two separate subsets: Ped-1 and Ped-2. Both of these subsets include sequences of pedestrians captured with stationary surveillance cameras. Consequently, the images in the dataset are black and white and relatively uniform.

The test sequences include instances of anomalous movement, such as skaters, cyclists, and service vehicles. While visual inliers like runners or cyclists may also perform anomalous movement, we find that most of the anomalous movement is associated with anomalous objects such as bicycles, skateboards, and other vehicles. Therefore, we employ this dataset to evaluate our approach to visual outlier detection in contexts other than road driving.

The Ped-1 dataset contains 34 train sequences and 36 test sequences, 10 of which have densely annotated anomalies. The Ped-2 dataset contains 16 training sequences and 12 test sequences.

5.1.8 StreetHazard

StreetHazard [114] is a synthetic dataset created using the UnrealEngine and the CARLA simulation environment to insert realistic anomalous objects into road-driving scenes. This approach offers a distinct advantage over other methods of synthetic mixed-content image creation, as it avoids issues such as inconsistent lighting or chromatic aberrations that might otherwise enable anomaly detection.

The authors of StreetHazard have employed three locations in the CARLA simulator to generate 5125 labelled training images with a 12-class taxonomy. They have used a fourth

location to create a validation dataset containing 1031 images. Additionally, they have leveraged the fifth and sixth locations to generate 1500 test images with outlier samples. These anomalies were created using 250 unique anomaly shapes from the Digimation Model Bank Library and semantic ShapeNet.

5.1.9 Additional validation datasets

One of the main potential drawbacks of our approach is that we both train and validate on synthetic mixed content images. This means that it is not not always clear whether the model has learned to properly detect outliers or if it is merely reacting to pasting artefacts and other simple cues that may indicate an outlier patch.

We therefore created two mixed-content validation datasets by pasting Pascal animals into Vistas validation images, with the two differing on the amount of preprocessing on the pasted patches. Unlike the WD-Pascal dataset, there is no domain shift between the training and validation inlier samples.

We further created three control datasets to ensure that our models do not simply react to pasted content.

We next describe these 5 sets.

PascalVistas10

We began by searching for Pascal images that contained segmentation ground-truth for any of the seven animal classes: bird, cat, cow, dog, horse, and sheep. From there, we selected 369 large Pascal objects from their original images using pixel-level segmentation ground-truth. To create the mixed-content validation dataset, we chose a random image from the Vistas validation set and resized the selected object to cover at least 10% of the image's pixels. We then pasted the object at a random location within the image. The result was 369 combined images.

PascalVistas1

A potential issue with resizing objects before pasting is that the outlier detection model may detect the pasted objects by recognizing the resizing artefacts, rather than the novelty. In order to address this issue, we form another dataset as follows. We iterate over all instances of Pascal objects and choose a random image from the Vistas validation dataset. We paste the object without any resizing only if it takes at least 1% image pixels. This results in 31 combined images. This datasets is more difficult than the previous one since outlier patches are much smaller.

Vistas to City

To test whether our approach is capable of detecting real outliers or simply reacting to differences in pasted textures, we created our first control set. This was done by pasting a random object instance from Vistas val into a random image from Cityscapes val. The pasted instance had to occupy at least 0.5% of the Cityscapes image, with no preprocessing performed before pasting.

Since both Vistas and Cityscapes datasets contain only inlier classes, the performance on this set is an indicator of whether the model can distinguish real outlier pixels from differences in camera characteristics. This dataset consists of 1543 images.

CityCity

We create the second control set by randomly pasting an object instance from Cityscapes validation dataset onto a different validation image from the same dataset. The only requirement is that the object instance should cover at least 0.5% of the image area. No preprocessing is applied to the patch prior to pasting.

Since both the patch and the image belong to the same dataset and should contain only inlier classes, the model's performance on this set reveals whether it can distinguish between inlier pixels and outlier pixels due to the differences in imaging conditions between the patch and the image.

This dataset comprises 288 images. The texture differences between the image and the pasted patch are expected to be smaller than in the case of the Vistas to City control set.

SelfSelf

We created the final control set by randomly selecting an object instance from a Vistas image and pasting it onto a different location in the same image. The object instance had to cover at least 0.5% of the Vistas image, and no preprocessing was applied before the pasting.

Since the pasted object is from the same image as the background, there should be no difference in texture or lighting. Performance on this set tests whether the model can detect objects in unexpected locations within the same scene.

The set consists of 1873 images.

5.2 Validation measures

We next describe the two measures that were used for estimating segmentation and outlier detection quality.

5.2.1 Mean intersection over union

Mean intersection over union (mIoU) is a common evaluation metric used in image segmentation tasks. It measures the similarity between the predicted segmentation mask and the ground truth mask by computing the intersection over union (IoU) for each class and then taking the average over all classes.

The Intersection over Union (IoU) is defined as the ratio between the area of intersection of the ground-truth mask G and the prediction mask P for a given class, and the area of their union (cf. Figure 5.1). The intersection of the ground-truth mask and the prediction mask is equal to the number of true positives (TP), while the union of the tow masks is equal to the sum of true positives, false positives (FP) and false negatives (FN). Therefore:

$$IoU = \frac{G \cap P}{G \cup P} = \frac{G \cap P}{G \cap P + G \setminus P + P \setminus G} = \frac{TP}{TP + FN + FP} \tag{5.1}$$

mIoU is a popular metric for evaluating the performance of segmentation models because it takes into account both false positives and false negatives and provides a single value that summarizes the overall accuracy of the model. Higher mIoU values indicate better segmentation performance.

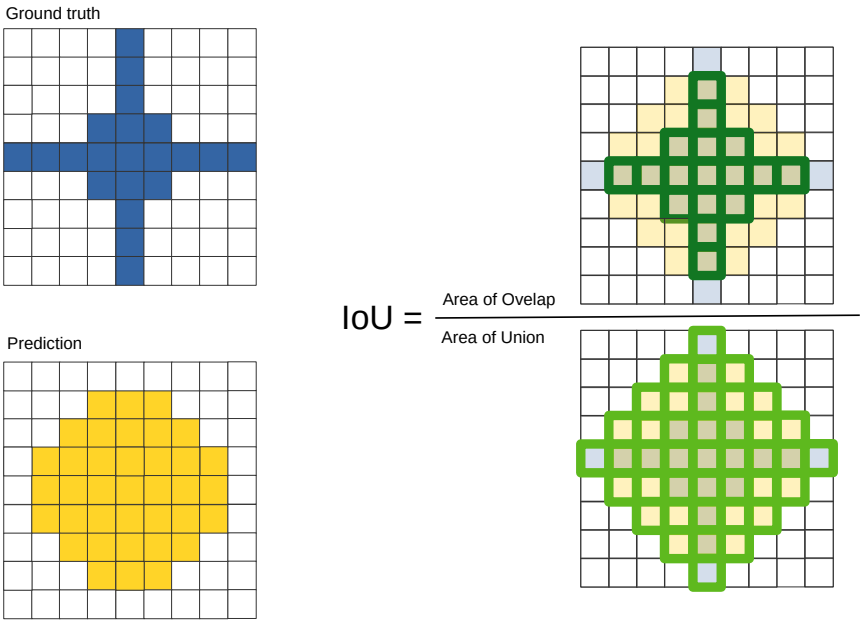

Figure 5.1: Intersection over Union is calculated by dividing the area of overlap between the ground-truth and the prediction mask by the area of their union

5.2.2 Average precision

Average precision (AP) is a performance metric for binary classification problems that measures the area under the precision-recall curve. It takes into account both the precision and recall of the model predictions over different thresholds of the predicted probabilities.

In binary classification, the precision is the ratio of true positives to the total number of predicted positives (TP + FP). Recall, on the other hand, is the ratio of true positives to the total number of actual positives (TP + FN).

The precision-recall curve is a plot of precision against recall at different thresholds of the predicted probabilities. The average precision is then calculated by computing the area under this curve. A higher average precision indicates a better-performing model.

We only measure AP for the outlier pixels.

5.3 Training details

Our models are primarily trained on inliers from the Vistas train dataset. However, we augment our training data for benchmark submissions by including images from the Cityscapes train dataset and WildDash validation set. Notably, we do not incorporate any validation data into our training process for submissions. For instance, we do not use Fishyscapes Lost and Found for submission training. For validation experiments on the CAOS, Vistas-NP, and UCSD datasets, we use the corresponding training datasets.

In most of our validation experiments, we use the ImageNet-1k-bb dataset as a negative dataset. To train our models, we use standalone negative images as well as mixed-content images, which are obtained by pasting a resized negative image into an inlier image. Initially, we resized each negative image so that its area became 5% of the inlier image. However, we later improved this procedure by randomly scaling the negative images to sizes between 0.1% and 10% of the inlier area.

We evaluate the quality of our models' outlier detection on WD-Pascal and WD-LSUN [17], as well as on Fishyscapes Lost and Found val, where we measure AP. To evaluate our segmentation quality, we measure mIoU on WildDash validation images. We also conduct experiments on our control sets to assess if our models respond to pasting cues. We evaluate our models on various benchmarks, including WildDash, Fishyscapes, CAOS, Vistas-NP, and a subset of UCSD anomaly test that has dense annotations available.

Our models are primarily based on the DenseNet-169 backbone [36] with ladder-style upsampling [37], as it has demonstrated the best overall validation performance. However, we make an exception for the discriminative models, where we use DenseNet-121 without upsampling to reduce capacity and discourage overfitting. Regardless of the backbone, we upsample the predictions to the input resolution using bilinear interpolation.

To prepare images for training, we first normalize them using the ImageNet mean and variance. During training, we form batches by randomly cropping the images to a size of 512x512 pixels and applying horizontal flipping for additional data augmentation. We do not use multi-scale evaluation. However, we have different strategies for rescaling images depending on whether we incorporate scale jittering into training. For images without scale jittering, we resize them so that the smaller side is 512 pixels before taking the crop. We do the same for input images during validation. When scale jittering is included during training, we randomly resize 70% of the images so that their smaller side falls between 512 and 1536 pixels, while the remaining 30% are resized according to the original protocol. During evaluation, we resize input images so that the smaller side is 768 pixels. These rescaling strategies improve the overall performance of our models.

We use the standard Adam optimizer and a learning rate which is decreased with a cosine learning policy from $4 \cdot 10^{-4}$ to $1 \cdot 10^{-7}$. We reduce the learning rate of pre-trained backbone parameters by a factor of 4 during training. Our models are trained for 75 Vistas epochs, which is equivalent to 5 epochs of ImageNet-1k-bb. However, we increased the number of training epochs to 20 for our benchmark submissions. During inference, we detect outliers by setting a threshold on the inlier probability, which we set to $p_{\text{IP}} = 0.5$. discussed in Section 4.3. We set the weights from Table 4.2 to λ_{AUX}=0.4, λ_{KL}=0.2, $\lambda_{\text{C+1}}$=0.05, and λ_{TH}=0.2. These hyper-parameters were validated in experiments conducted on WD-Pascal and WD-LSUN datasets [98].

5.4 Baseline dense anomaly detection

Table 5.1 shows the results of image-wide outlier detection approaches adapted for dense outlier detection.

We train three models: a standard C-way multi-class model, a C-way multi-class model with MC-dropout and a model with a confidence prediction head.

The first two rows of the table illustrate the performance of the C-way multi-class model in outlier detection, with max-softmax used as the criterion for outlier detection. The first row represents the baseline performance of the model, while the second row shows the performance of the model with ODIN [67]. ODIN is a pre-processing method that increases the winning softmax score by perturbing the input. The assumption is that in-distribution samples should be more strongly affected than out-of-distribution samples, thereby making them more separable. Our experimental results demonstrate that the use of ODIN results in a slight improvement in performance across all experiments.

The third row shows the multi-class model trained using MC-dropout which we apply to the output of each dense layer and upsample block. We set the dropout rate to 0.2. For outlier

detection, we use epistemic uncertainty, which we calculate over 50 forward passes as a difference between ensemble entropy and the mean value of individual output entropies. This setup achieves the best outlier detection results, at the expense of slightly deteriorated segmentation accuracy.

Table 5.1: Validation of anomaly detection approaches adapted for dense prediction. WD denotes WildDash 1 val, MC denotes models trained and evaluated using MC-dropout with 50 forward passes.

Model	AP WD-LSUN	AP WD-Pascal	mIoU WD
C× multi-class	55.6 ± 0.8	5.0 ± 0.5	50.6
C× multi-class, ODIN	56.0 ± 0.8	6.0 ± 0.5	**51.4**
C× multi-class, MC	$\mathbf{64.1 \pm 1.0}$	$\mathbf{9.8 \pm 1.2}$	48.4
confidence head	54.4 ± 0.8	3.4 ± 0.4	46.4

The last row of the table presents an interesting approach to address the challenge of recognizing difficult samples, which involves using a separate confidence head trained with a fully convolutional variant of the protocol proposed by [73]. This head estimates the confidence of the model's predictions, and the original predictions are then adjusted by interpolating between them and the target probabilities, based on the confidence score: $P'(Y_{ij} = y_{ij}|x) = c_{ij}P(Y_{ij} = y_{ij}|x) + (1 - c_{ij})y_{ij}$. The adjusted probabilities are then used to compute the $\mathscr{L}_{\text{MC}}$ loss. Additionally, the model is encouraged to have high confidence in all predictions through a second loss term $\mathscr{L}_{\text{C}}$. Thus, lowering the confidence increases $\mathscr{L}_{\text{C}}$ but decreases $\mathscr{L}_{\text{MC}}$, creating a game-like situation where the model learns to recognize difficult samples where it benefits from higher uncertainty. Still, the model with a separate confidence head achieves the lowest overall score across all tasks.

Overall, all of the adapted approaches can be said to have a low outlier detection accuracy. Figure 5.2 shows the qualitative results for the baseline model on WD-Pascal. It illustrates the two main factors that influence the low score. The first is that the model tends to have highly confident predictions in unknown pixels. The second is that the model tends to have high uncertainty on semantic borders.

5.5 Discriminative anomaly detection

In this section, we evaluate the performance of discriminative outlier detectors. We formulate the detector as binary classification over two classes: inlier and outlier.

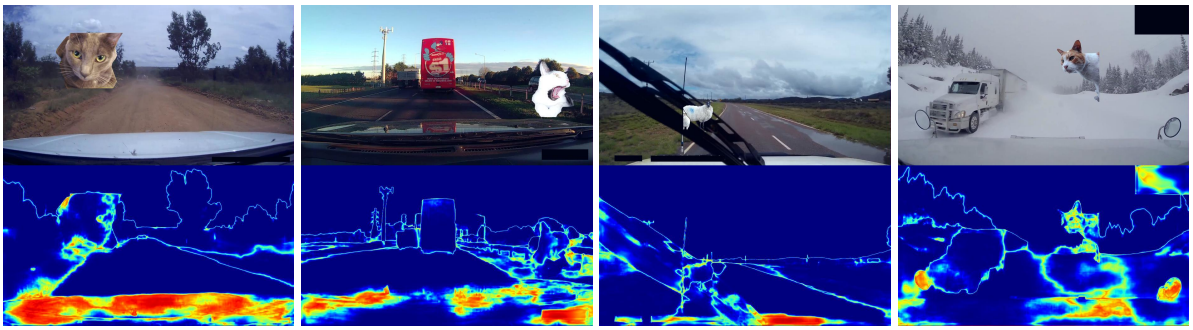

Figure 5.2: Dense outlier detection for the baseline model presented in Table 5.1. Row 1 shows WD-Pascal images. Row 2 shows outlier probabilities obtained with the C-way multi-class model. Red colour indicates a high probability that a pixel is an outlier. Models that rely on uncertainty estimation tend to have high confidence on unknown input and high uncertainty on semantic borders.

5.5.1 Road-driving images

We first apply our discriminative detector in road-driving context. We trained two variants of the detector: one that uses only negative images during training and another that further includes pasted negative patches. Table 5.2 shows the performance of both models on four datasets: WD-LSUN, WD-Pascal, PascalVistas10, and PascalVistas1. The results indicate that including pasted patches in training significantly improves the performance of the outlier detector, especially on mixed-content images. We observed that the difference in performance between WD-Pascal and PascalVistas* datasets may be due to domain shift between Vistas and WildDash, which affects the quality of inlier detection. Furthermore, the model performs best on PascalVistas10, indicating that it works better on larger outlier patches.

Table 5.2: Average precision for discriminative OOD detection on the validation datasets. The no-pasting model was trained on full negative images, while the pasting model was trained on both negative and mixed-content images.

Model	WD-Lsun	WD-Pascal	PascalVistas10	PascalVistas1
no pasting	98.9 ± 0.0	2.4 ± 0.3	13.1	2.4
pasting	98.5 ± 0.1	7.9 ± 2.5	**87.9**	**78.6**

Figure 5.3 shows the qualitative results for the two considered models. The results further demonstrate the model trained without pasting is virtually unable to detect the pasted patches. Notice that the discriminative model has no problem identifying pixels at semantic borders as inliers.

We now turn to the performance of the two models on the three control datasets, and present the results in Table 5.3. The models achieve the highest AP score on Cityscapes images with pasted Vistas patches (VistasCity), indicating that the model can react to pasted textures and imaging conditions. However, the average precision for control datasets is still significantly

Figure 5.3: OOD detection in Vistas images with pasted Pascal animals. Row 1 shows a validation image from PascalVistas1, while row 2 presents an image from PascalVistas10. The columns correspond to: i) original image, ii) discriminative OOD detection trained without pasting and ii) discriminative OOD detection trained with pasting

lower than for some of the validation datasets, suggesting that the model also reacts to the semantics of pasted patches. Overall, these results demonstrate that the discriminative outlier detector can effectively detect outliers in mixed-content images, but its performance may be affected by the specific domain and content of the images.

Table 5.3: AP for detection of pasted content in the three control datasets for the two variants of the discriminative model.

Model	CityCity	VistasCity	SelfSelf
no pasting	2.4	9.1	3.6
pasting	7.6	34.1	19.7

We show the qualitative results on control datasets in Figure 5.4. Compared to results on the validation dataset, the response on pasted patches is generally lower, with most of the images having low outlier probability.

5.5.2 UCSD anomaly dataset

We demonstrate the effectiveness of our approach for dense novelty detection on the Peds1 and Peds2 subsets from the UCSD anomaly dataset, which consists of real-world mixed-content images captured in non-road-driving contexts. To evaluate our approach, we use only the test

Figure 5.4: Examples of detection of pasted content in three control datasets: Cityscapes with pasted Vistas content (row 1), Cityscapes with content pasted from other Cityscapes images (row 2) and Cityscapes with content pasted from the same image (row 3).

sequences with dense ground truth annotations, which include all test sequences from Peds2 and 10 sequences from Peds1 (S3, S4, S14, S18, S19, S21-S24, S32).

It should be noted that this experimental setup involves relatively uniform training data and a weaker learning signal as there are no semantic segmentation labels. We only train the discriminative outlier detector and use ImageNet1k-bb as a negative data source. To validate the backbone capacity and the usefulness of the upsampling path, we begin by using sequence 1 of UCSD Peds2. The results presented in Table 5.4 support our hypothesis that this dataset is not suitable for models with excessive capacity. As a result, we conduct the remaining experiments using a DenseNet-121 backbone without the upsampling path.

Table 5.4: Validation experiments on sequence S1 from the UCSD Peds2 test dataset. The models were trained on the UCSD dataset (inliers) and ImageNet-1k-bb (outliers).

DenseNet depth	Upsampling	AP	AUROC
169	✓	6.8	60.8
169	✗	20.9	**91.3**
121	✓	28.7	85.2
121	✗	**46.7**	88.7

Table 5.5 shows our frame-level AUROC (Area Under Receiver Operating Characteristic curve) results on the UCSD dataset, which is a common metric used in anomaly detection. The

ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at different thresholds. A higher AUROC score indicates better performance and enables us to compare our model to previous work.

Table 5.5: AUROC per frame on the Ped 1 dataset. Our model is based on DenseNet-121, does not use ladder upsampling and was trained on the UCSD dataset (inliers) and ImageNet-1k-bb (outliers).

Model	AUROC
Sugiyama et al. [115]	67.5
Ionescu et al. [116]	82.2
Pang et al. [117]	83.2
Liu et al. [118]	87.5
Liu et al. [55]	95.4
Nguyen et al. [53]	96.2
Park et al. [54]	97.0
Ionescu et al. [52]	97.8
Discriminative model (ours)	91.2

Our approach does not achieve state-of-the-art results as it does not consider temporal information. This information is especially useful in the context of anomalous movement detection which is the primary focus of the UCSD dataset. Furthermore, our approach is designed to detect visual anomalies, which do not correspond to anomalous movement in all cases. Still, our performance indicates that a simple discriminative outlier detector achieves good results even in a highly specialized setup such as UCSD without any modifications to our approach.

To provide a deeper insight into the differences between visual and movement anomalies, we examine the results on the remaining sequences from Peds2, as well as on some individual sequences, presented in Table 5.6. The model achieves its best performance on sequence S4, which includes a small utility vehicle that is also a visual outlier. The model also performs well on sequence S8, which features bicycles. This sequence has a lower FPR95 score because the model does not identify the rider as anomalous, but rather the wheels of the bike, which are visually anomalous. In sequence S7, which contains two bicycles and a skater, one of the bikes is not labelled as anomalous since it was pushed and not ridden. In sequence 12, which includes a skater, our model missed the skateboard, which is relatively small.

Table 5.7 shows accuracy on Peds1 test. Peds1 is more challenging than Peds2 due to the prominent depth of the scene and lower position of the camera. Our model detects carts in sequences 14 and 19, but is unable to detect the skater in sequence 18, the person in the

Table 5.6: AP for OOD detection on UCSD Peds2 sequences 2-10 combined and 4, 7, 8 and 12 individually denoted as S 2-10, S4, S7, S8 and S12 respectively. Our model is based on DenseNet-121, does not use ladder upsampling and was trained on the UCSD dataset (inliers) and ImageNet-1k-bb (outliers).

	S2-S10	S4	S7	S8	S12
AP	54.5	90.2	10.3	50.6	2.1
AUROC	89.6	99.1	85.6	79.5	65.6

wheelchair in sequence 23, and the bicyclist in sequence 32. We believe that the last false negative occurs because the wheels of the bike are not visible.

Table 5.7: AP for OOD detection on the whole UCSD Peds1 test dataset, as well as on sequences S14, S19, S18, S32, and S23. Our model is based on DenseNet-121, does not use ladder upsampling and was trained on the UCSD dataset (inliers) and ImageNet-1k-bb (outliers).

	all	S14	S19	S18	S32	S23
AP	44.3	64.8	87.6	0.2	2.6	0.7
AUROC	81.8	89.7	99.8	65.3	79.2	56.2

Figure 5.5 shows some qualitative results on both UCSD subsets. The first column represents the input image the second the ground truth with anomalies marked in white, and the third the probability that a pixel is an outlier. Our model confidently detects the small utility vehicle (rows 2 and 5) and bicycle wheels (rows 1, 2 and 4), but it misses cyclists when parts of the bike are not visible (rows 4 and 5). Our model also detects transmission noise though this noise is not marked as an anomaly in the ground truth image (row 6).

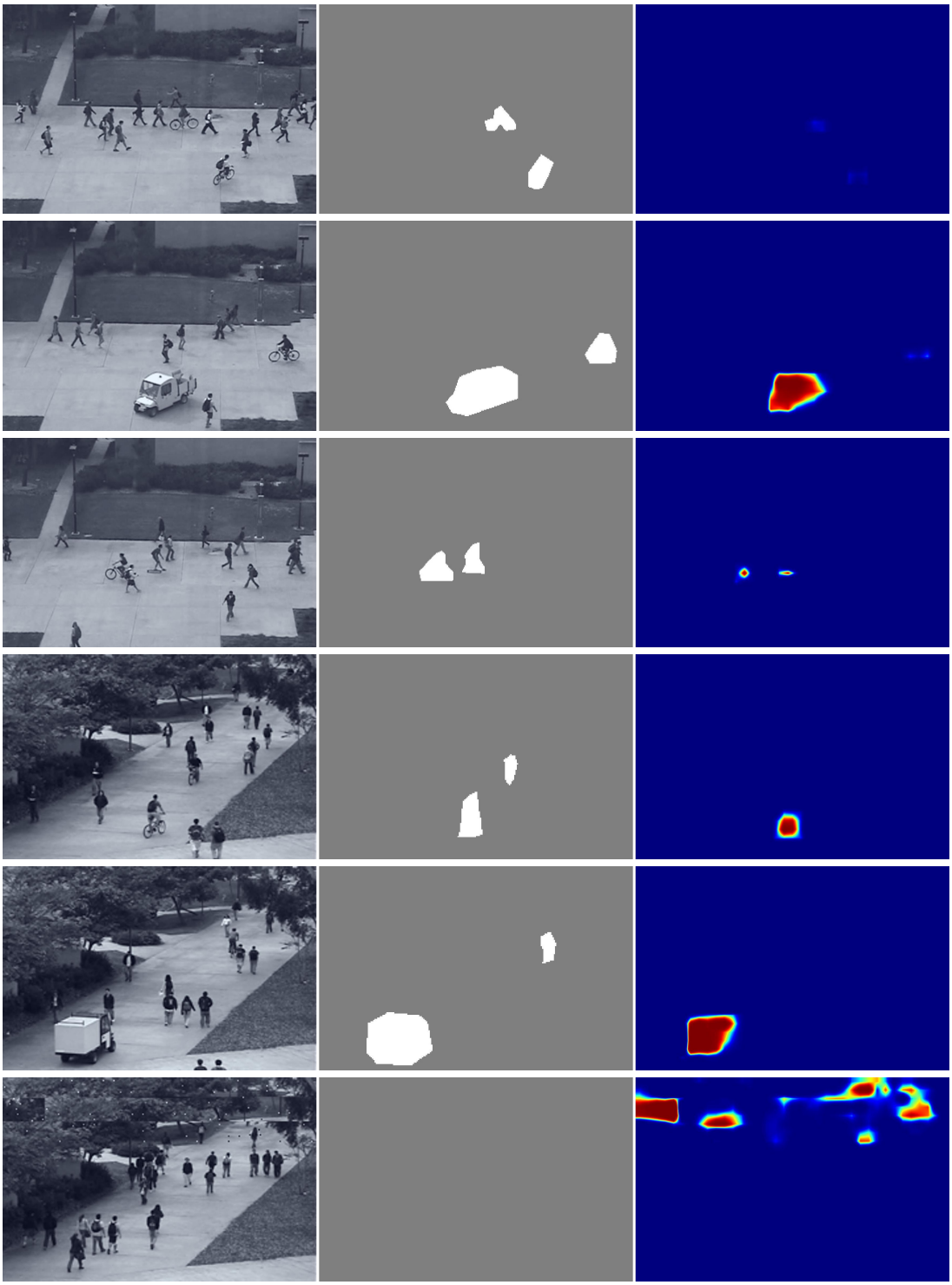

Figure 5.5: Dense outlier detection results on UCSD Peds 2 (rows 1-3) and UCSD Peds 1 (rows 4-6). Column 1 shows the original image, while columns 2 and 3 contain the ground-truth and our predictions, respectively. Our method identifies visual anomalies such as bike wheels, though it does not identify bike riders as outliers (rows 1,3 and 4). The method performs better on larger outliers such as the cart (row 2 and 5). We even detect image quality failures as visual anomalies (row 6).

5.6 Dense open-set recognition

The following section considers our open-set recognition approaches, as described in Section 4.3.

5.6.1 Validation of open-set recognition modules

Table 5.8 compares the four different open-set recognition modules which were trained using noisy negatives. The results on WD-LSUN and WD-Pascal, compared to those in Table 5.1, show that training with noisy and diverse negatives has significantly improved outlier detection. However, we have observed a reduction in the segmentation score. This reduction is least pronounced for the C-way multi-class model and the two-head model, which we will analyze next. In addition, the open-set modules perform better on WD-Pascal than pure discriminative models (cf. Table 5.2). This indicates that outlier detection improves when combined with semantic segmentation.

Upon analyzing the individual results, we observed that the two-head model has a slight disadvantage in discriminating WildDash val from LSUN compared to the single-head C-way approach, as it is more sensitive to domain shift between Vistas train and WildDash val. However, the two-head model outperforms the single-head approach in terms of inlier segmentation by 0.7 percentage points in column 4, and outlier detection on Pascal animals by 5 percentage points in column 3. Our qualitative analysis revealed that these advantages stem from the fact that the single-head C-way approach generates numerous false positive outlier detections at semantic borders due to a low max-softmax score.

The C+1-way multi-class model performs the worst out of all models trained with noisy outliers. The multi-label model performs well on outlier detection but significantly worse on inlier segmentation.

Table 5.8: Validation of dense open-set segmentation approaches. All of the models were trained with on the Vistas dataset and used ImageNet-1k as noisy negatives. The models saw both negative and mixed content images during training.

Model	AP WD-LSUN	AP WD-Pascal	mIoU WD
two heads(=LDN_BIN)	99.3 ± 0.0	34.9 ± 6.8	**47.9**
C× multi-class(=LDN_OE)	$\mathbf{99.5 \pm 0.0}$	33.8 ± 5.1	47.8
C+1× multi-class	98.9 ± 0.1	25.6 ± 5.5	46.2
C× multi-label	98.8 ± 0.1	$\mathbf{49.1 \pm 5.6}$	43.4

In Figure 5.6, we provide a qualitative analysis of the performance of different open-set modules. The first row shows images from WD-Pascal, while the subsequent rows show the

output of different outlier detectors. The C+1-way multi-class model, shown in the second row, achieves the lowest AP score due to its tendency to make more false outlier predictions. The third and fourth rows show the C-way multi-label and C-way multi-class models, respectively. These models perform similarly and are better at outlier detection than the baseline max-softmax approach. However, they also tend to assign a high outlier probability to border pixels. The two-head model, shown in the last row, successfully detects outliers without falsely detecting borders as outliers. Moreover, its detections are coarser than those of the other models. Overall, the two-head model appears to be the most effective at minimizing false detections at the borders.

All of the trained models seem to struggle with a significant domain shift, as shown in column 4. This suggests that achieving high AP scores on WD-Pascal would be difficult, as images with significant domain shifts are perceived as equally anomalous as pasted Pascal objects. In column 3, models trained with negative data classify the windshield wiper as an outlier. This is noteworthy when compared to the model that has seen WildDash validation during training, as shown in row 3 of Figure 5.15. Further discussion of this is provided in later sections.

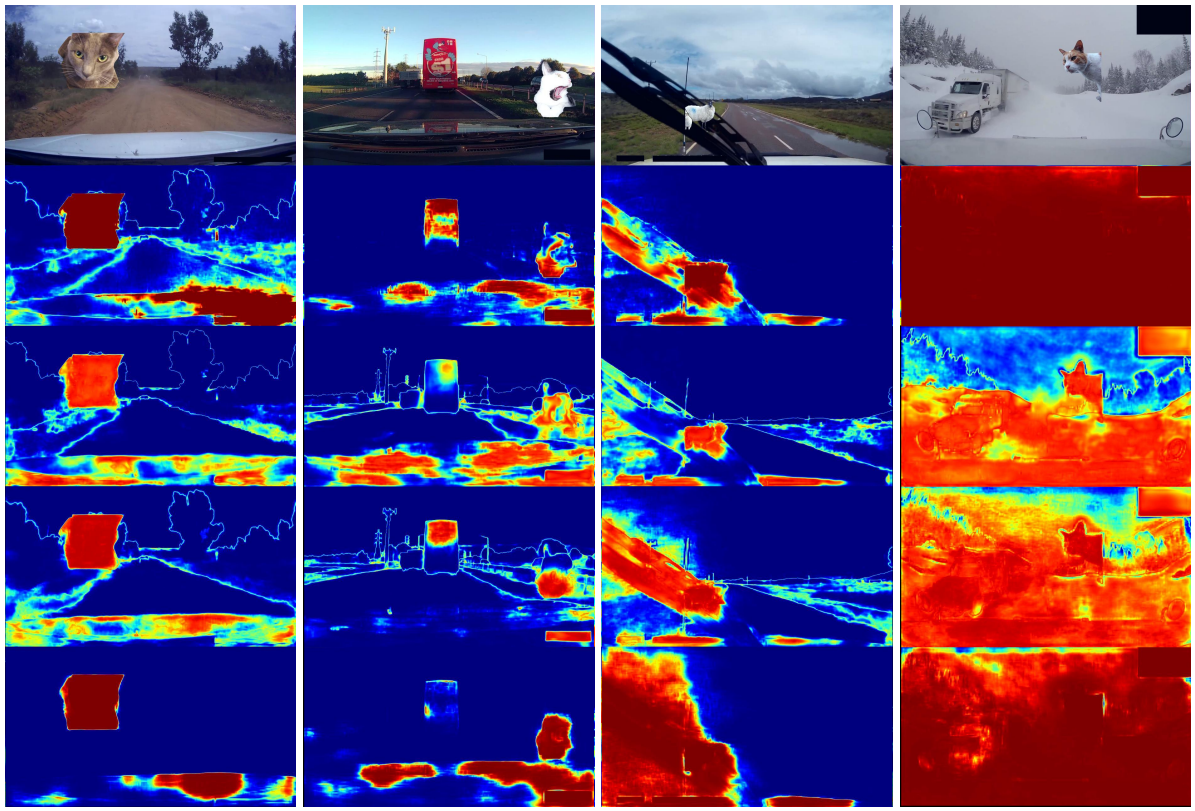

Figure 5.6: Dense outlier detection with models presented in Table 2. Row 1 shows Wilddash val images with pasted PASCAL VOC 2007 animals. Subsequent rows correspond to models trained with noisy negatives: the C+1-way multi-class model (row 2), the C-way multi-label model (row 3), the C-way multi-class model (row 4) and the two-head model (row 5). Red colour indicates a high probability that a pixel is an outlier.

5.6.2 Validation of Dense Feature Extractor Backbones

Table 5.9 explores influence of different backbones to the performance of our two-head model. We experiment with ResNets and DenseNets of varying depths. The upsampling blocks are connected with the first three DenseNet blocks. In the ResNet case, the upsampling blocks are connected with the last addition at the corresponding subsampling level.

Table 5.9: Validation of backbones for the two-head model. WD denotes WildDash val. All of the models were trained with on the Vistas dataset and used ImageNet-1k as noisy negatives. The models saw both negative and mixed content images during training.

Backbone	AP WD-LSUN	AP WD-Pascal	mIoU WD
DenseNet-121	99.1 ± 0.0	41.4 ± 7.0	44.8
DenseNet-169	$\mathbf{99.3 \pm 0.0}$	35.7 ± 5.9	47.4
DenseNet-201	98.3 ± 0.1	27.2 ± 5.7	**47.6**
ResNet-34	97.2 ± 0.1	37.1 ± 5.6	45.2
ResNet-50	99.1 ± 0.0	37.8 ± 5.6	41.7
ResNet-101	99.0 ± 0.1	36.9 ± 5.1	43.7

All models achieve very good outlier detection in negative images. There appears to be a trade-off between outlier detection and semantic segmentation accuracy.

5.6.3 Validation of the Training Datasets

Table 5.10 explores the influence of inlier training data to the model performance.

Table 5.10: Influence of the inlier training dataset to the performance of the two-head model with the DenseNet-169 backbone. WD denotes WildDash val. All of the models used ImageNet-1k-bb as noisy negatives. The models saw both negative and mixed content images during training.

Inlier training dataset	AP WD-LSUN	AP WD-Pascal	mIoU WD
Cityscapes	66.6 ± 0.9	12.6 ± 1.8	11.1
Vistas	99.3 ± 0.04	35.7 ± 5.9	47.2
Cityscapes, Vistas	$\mathbf{99.3 \pm 0.0}$	$\mathbf{39.1 \pm 6.3}$	**47.8**

The results suggest that there is a very large domain shift between Cityscapes and WildDash val. Training on inliers from Cityscapes leads to very low AP scores, which indicates that many WildDash val pixels are predicted as outliers with respect to Cityscapes. This suggests that Cityscapes is not an appropriate training dataset for real-world applications. Training on inliers from Vistas leads to much better results which is likely due to the greater variety with respect

to the camera, time of day, weather, resolution etc. The best results across the board have been achieved when both inlier datasets are used for training.

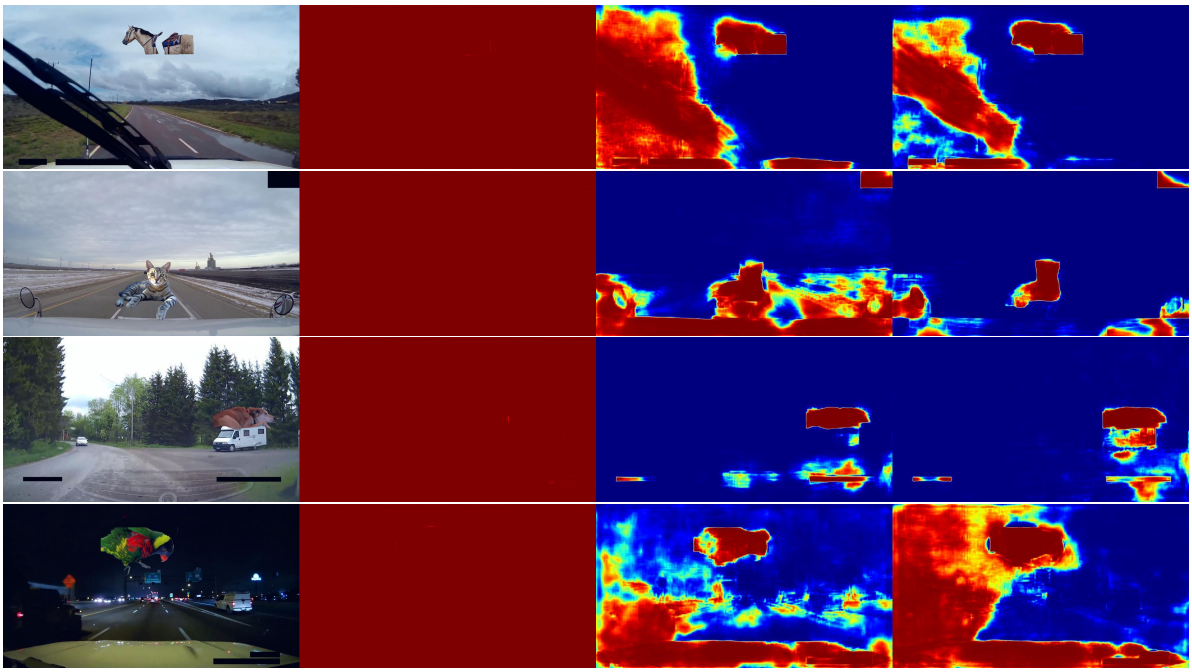

Figure 5.7: Outlier detection with two-head models trained on different inlier datasets. All models have been trained with pasted noisy negatives from ImageNet-1k-bb as presented in Table 4. Column 1 contains Wilddash images with pasted PASCAL VOC 2007 animals. Columns 2-4 show predictions of models trained on Cityscapes, Vistas, and Cityscapes and Vistas, respectively. Red colour indicates a high probability that a pixel is an outlier.

Table 5.11: Influence of the outlier training dataset on the performance of our two-head model with the DenseNet-169 backbone. WD denotes WildDash val. All of the models were trained with on the Vistas dataset. The models saw both negative and mixed content images during training.

Outlier training dataset	AP WD-Pascal	mIoU WD
ImageNet-1k-full	35.8 ± 6.9	43.7
ImageNet-1k-bb	35.7 ± 5.9	47.2
ADE20k	15.7 ± 2.2	**49.9**

Table 5.11 explores the impact of negative training data. The table shows training on ImageNet-1k-bb and ADE20k significantly boosts inlier segmentation when compared to ImageNet-1k. We hypothesize that this occurs because ImageNet-1k-bb has a smaller overlap with respect to the inlier training data. This simplifies outlier detection due to decreased noise in the training set, and allows more capacity of the shared feature extractor to be used for the segmentation task. ADE20k dataset seems to be easily distinguished from the Vistas which reduces the regularization effect of the outlier detection head and leads to better segmentation performance. On the other hand, this overfitting to the inlier training data leads to a reduced outlier detection

performance on WD-Pascal. Furthermore, as ImageNet-1k-bb contains a lot of animal classes, it may also be closer to Pascal outliers. The table omits outlier detection in negative images, since all models achieve over 99% AP on that task.

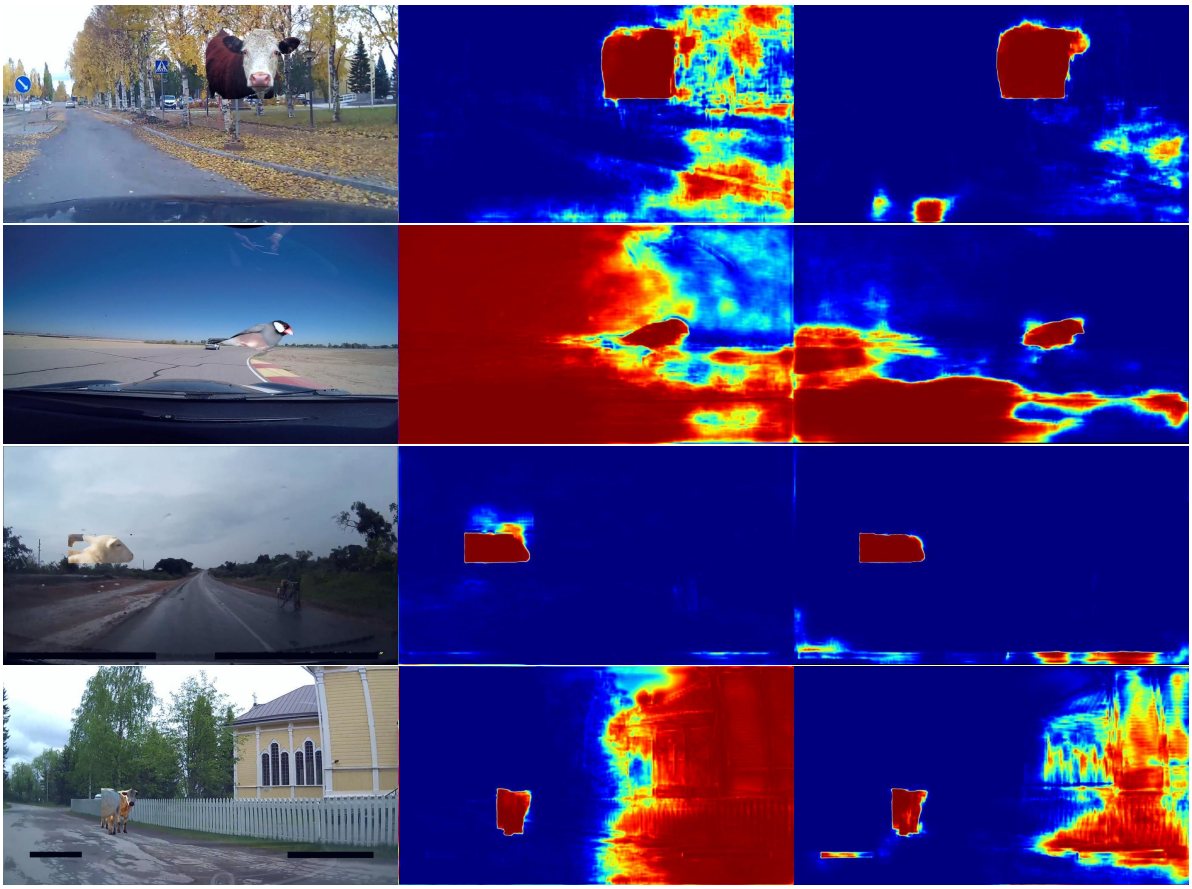

Figure 5.8: Outlier detection with two-head models trained on different negative datasets. All models have been trained by pasting negatives into inliers from Vistas as presented in Table 5. Column 1 shows the original Wilddash images with pasted PASCAL VOC 2007 animals. Columns 2 and 3 show predictions of models trained with ImageNet-1k-full and ImageNet-1k-bb, respectively. Red colour indicates a high probability that a pixel is an outlier.

Figure 5.7 illustrates the validation performance of the two-head model depending on the inlier training dataset. Column 1 shows four validation images. Column 2 presents the corresponding results of the two-head model trained on Cityscapes. This model classifies all of the WildDash pixels as outliers. This indicates that models trained on Cityscapes are very sensitive to domain shift. Column 3 shows that the two-head model trained on Vistas dataset is significantly better at outlier detection. This improvement indicates that models trained on Vistas show more resilience to domain shift. Column 4 depicts a model trained on both Vistas and Cityscapes. It performs similarly to the model trained on Vistas. Interestingly, the model trained on Vistas shows more resilience to unusual conditions (dark image in row 4, unusual vehicle in row 3), while the model trained on combined datasets shows more precision in normal situations (grass in row 1, road in row 2).

Figure 5.8 shows the results of the two-head model depending on the negative training dataset (cf. Table 5). The first column shows the validation images. The second column illustrates the model trained using ImageNet-1k-full with pasting. It is capable of detecting outlier samples. It is however sensitive to domain shift. This is because of the increased overlap between positive and negative images (in classes such as sky, vegetation or road, which appear often in the backgrounds of ImageNet images). The last column shows the model trained using ImageNet-1k-bb with pasting which performs the best both qualitatively and quantitatively.

5.6.4 Validation of training augmentations

We next validate training augmentations, this time on the 100 publicly available Fishyscapes Lost and Found images. Table 5.12 shows the validation results. All models use Vistas and Cityscapes train and WildDash validation as inlier images. We use the outlier class activations for anomaly detection. We evaluate segmentation accuracy on Vistas validation dataset with Cityscapes classes using mIoU. We investigate the influence of three augmentations. We consider scale jittering (JS) where we vary image resolution before cropping it for training. We also explore pasting randomly scaled patches (RSP) where we vary the size of the patch before pasting it. Finally, we again look into training on pasted instances from ADE20k (Instances) where the pasted patches vary in shape and are not exclusively rectangular.

Table 5.12: Comparison of open-set segmentation approaches on Fishyscapes Lost and Found (AP) and Vistas (mIoU) validation subsets. We evaluate the contribution of scale jittering (JS), pasting of randomly sized crops (RSP) and use of object instances from ADE20K negative examples (Instances).

JS	RSP	Instances	AP L&F	mIoU Vistas
✗	✗	✗	13.2	75.1
✓	✗	✗	25.4	76.5
✓	✓	✗	36.9	76.3
✓	✓	✓	**50.3**	**76.7**

Both anomaly detection and semantic segmentation benefit from scale jittering during training. We further see that random scaling of negative samples before pasting improves the overall outlier detection performance. We speculate that this is primarily because of improvement on small outlier samples. Finally, though training on ADE20k worked poorly on WD-Pascal, it significantly improves outlier detection on Fishyscapes Lost and Found. This improvement likely comes from the fact that pasted patches have irregular shapes, unlike training on ImageNet-1k where we always paste squared patches. Pasting irregular shapes improves the precision of our outlier detector. Note also, that the domain gap between training and validation inlier data in

this experiment (Cityscapes vs. Lost and Found) is significantly smaller when compared to the experiment from the previous section (Vistas vs WildDash) which might further explain the difference in the results.

Figure 5.9 shows the performance of our model on Fishyscapes Lost and Found. Column 1 presents the original image. Column 2 contains the ground truth, with inlier, outlier and ignore pixels denoted in gray, white and black respectively. Finally, column 3 shows the output of our outlier detector. Our model performs very well on larger and closer objects (images 1-3), while struggling with distant and small objects. Notice that objects in the last image are so distant that it would be challenging even for a person to correctly classify them as an anomaly. It can be seen that our model classifies some of the ignore pixels (e.g. hood of the car and noise on image borders) as anomalies.

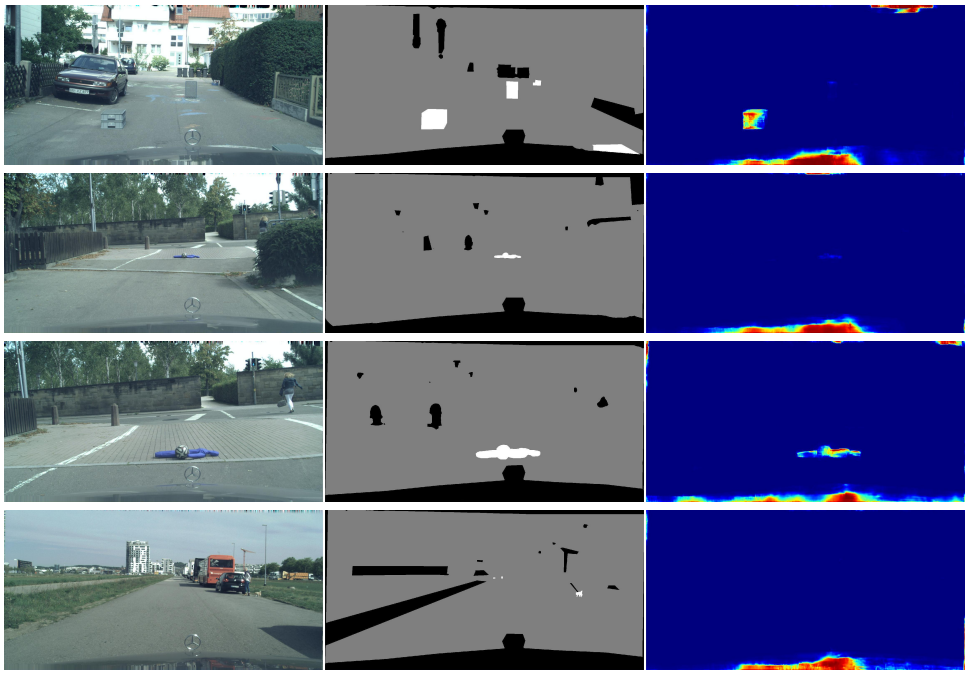

Figure 5.9: Results of the two head model trained with scale jittering and randomly sized patches on the publicly available Fishyscapes Lost and Found images. The original image can be seen in the first column, while columns 2 and 3 contain the ground truth and the outlier probability respectively. The model works better on closer objects than on distant ones (row 1). The confidence in outlier detection grows as an object draws near (rows 2 and 3). The model does not detect very small outliers (row 4).

5.6.5 Results on StreetHazards

Table 5.13 presents open-set segmentation accuracy on StreetHazard. We evaluate the two-head model trained with different augmentations and compare them with the max-softmax baseline. We use the outlier probability (OP) as predicted by the discriminative head for outlier detection. We ignore outlier pixels when measuring segmentation accuracy.

Unlike [103], we do not use ignore pixels during evaluation (same as [13]). Furthermore,

we do not report the mean of per-image anomaly detection scores. In our view, such practice may yield over-optimistic estimate of the overall anomaly detection metrics, since recognition errors can not propagate across images (e.g. these results will not reflect if the model is able to correctly identify negative images). We therefore determine global scores on 10 times down-sampled predictions. We evaluated the performance by measuring the mean of per-image scores and obtained similar results to the ones we report.

Table 5.13: Performance evaluation on StreetHazard. Our models were trained on StreetHazard (inliers) and ImageNet-1k-bb (negatives) and use the scale jittering augmentation. The C-was multi-class model is the baseline model trained without negative data and uses max-softmax (MSM) for outlier detection. We train two variants of the two-head model: one that uses fixed-size pasted patches (FSP) in mixed-content images and one with randomly sized patches (RSP). These models use outlier probability (OP) output of the outlier detection head for out-of-distribution detection.

Model	Negative data	Criterion	AP	test mIoU
PSPNet [103]	N/A	CRF+MSM	6.5	N/A
PSPNet [119]	N/A	TRADI	7.2	N/A
SPADE [97]	N/A	SynthCP	9.3	N/A
C-way multi-class	✗	MSM	7.3	65.0
two-head	FSP	OP	18.6	66.3
two-head	RSP	OP	**19.7**	**66.9**

Overall, our approach achieves better outlier detection performance than previously proposed approaches. The inclusion of negative data into training improves outlier detection. The best performing model has seen randomly sized negative patches in mixed-content images, which conforms to previous results.

Figure 5.10 shows the qualitative results. The columns represent: i) the original image, ii) the ground truth and iii) our output. Our model is able to detect larger outliers. It struggles with smaller outliers at the periphery of the image (row 3). Note that there is no overlap between outliers in our negative datasets and the synthetic outliers in the StreetHazards datasets. This suggests that our approach may generalize to outliers not seen in the negative dataset.

5.6.6 Results on Vistas-NP

We next perform experiments on Vistas-NP and show the results in Table 5.14. Interestingly, our approach yields worse results than the baseline max-softmax score. This is most likely due to the fact that the average size of outliers in Vistas-NP is 0.66% while the median size is 0.14%. It is not a surprise that our convolutional models underachieve on small objects. Our predictions

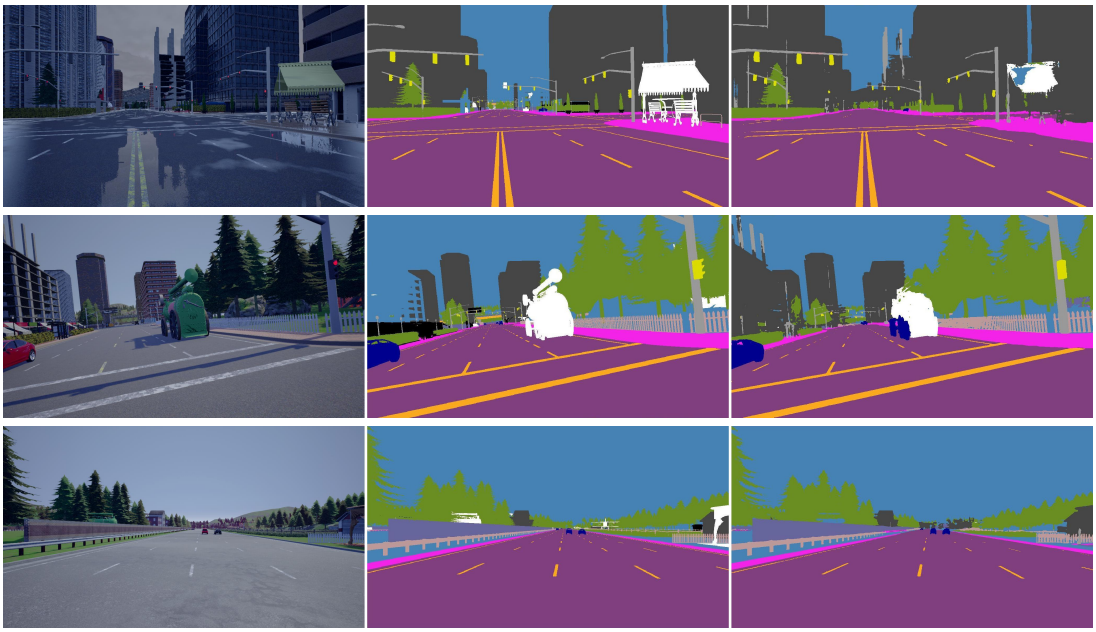

Figure 5.10: Open-set segmentation on StreetHazard. The first column is the original image, the second the ground truth, and the third is the output of the two-head model that was trained with randomly sized pasted patches. Outliers are white while ignore pixels are black. Our model performs better on large outliers (rows 1, 2) than on small ones (row 3).

are 4 times subsampled with respect to the input resolution in order to reduce computational complexity and memory footprint during training. This is a common trade-off [31] which can be avoided in principle, however at a great computational cost [120].

Table 5.14: Results of our models on Vistas-NP tes. Our models were trained on Vistas-NP train which does not contain any people (inliers) and ImageNet-1k-bb (negatives) and use the scale jittering augmentation. The C-was multi-class model is the baseline model trained without negative data and uses max-softmax (MSM) for outlier detection. We train two variants of the two-head model: one that uses fixed-size pasted patches (FSP) in mixed-content images and one with randomly sized patches (RSP). These models use outlier probability (OP) output of the outlier detection head for out-of-distribution detection.

Model	Negative data	Criterion	AP	test mIoU
C-way multi-class	✗	MSM	**12.4**	66.4
two-head	FSP	OP	10.9	**66.7**
two-head	RSP	OP	10.8	66.5

Figure 5.11 shows qualitative results for the two-head model that was trained with the randomly sized pasted patches. The columns represent: i) the original image, ii) the ground truth and iii) our output. The results clearly show that we are able to detect larger outliers closer to the camera, but struggle with distant and small objects.

Figure 5.11: Open-set segmentation on Vistas-NP. The first column is the original image, the second the ground truth, and the third is the output of the two-head model that was trained with randomly sized pasted patches. Outliers are white while ignore pixels are black.

5.6.7 Improving performance on small anomalies

The previous sections indicate that separate outlier detection head performs slightly worse on smaller outliers. We, therefore, look into potential improvements. In addition to AP, it is also useful to look at false positive rate at false negative rate of 95% (FPR 95). This measure calculates the rate of false positive predictions when we set the threshold so that the outlier detector identifies 95% of outliers. Lower FPR 95 score indicates better performance. Improvements in performance on small outliers will be more obvious on the FPR score than on the AP score.

We start by exploring the influence of outlier size on model performance by measuring correlation between the outlier area and the detection performance as shown in Figure 5.12. The figure shows AP and FPR 95 with respect to the area of the outlier patch for the baseline model which uses max-softmax for outlier detection (denoted with LDN_{JS}), and the two-head model that uses the output of the outlier detection head for outlier detection (denoted with $\text{LDN\_BIN}_{\text{JS, RSP}}$. We see that the accuracy of both models depend on outlier patch size. Max-softmax acts as an edge detector and therefore performs better on smaller objects. It however performs poorly on larger objects because it is unable to detect the interior of an object as an outlier.

Figure 5.12 implies that we can improve the accuracy of our two head models on small objects by combining the outlier probability with max-softmax. There are a few ways in which this can be done.

Firstly, assuming O is the output of the anomaly detection head, and i and j are the coordi-

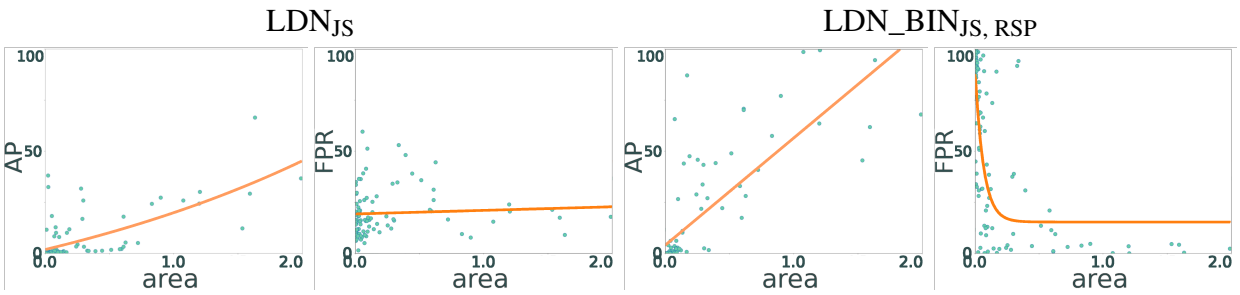

Figure 5.12: Influence of the outlier size on the model performance on Fishyscapes Lost and Found validation. Images 1 and 2 show AP and FPR95 using max-softmax baseline (LDN_{JS}) and images 3 and 4 show AP and FPR95 for our two-head model trained with noisy negatives ($\text{LDN\_BIN}_{\text{JS, RSP}}$). The orange line indicates an exponential function that was fitted to the data using least-squares. Higher AP and lower FPR scores indicate that our model prevails on large outliers. Max-softmax on the other hand achieves better results on small outliers because it detects object edges well.

nates of the pixel, we can use total probability:

$$\begin{aligned} P(\text{outlier}_{ij}|x) &= P(\text{outlier}_{ij}|O_{ij}=1)\cdot P(O_{ij}=1|x) \\ &+ P(\text{outlier}_{ij}|O_{ij}=0)\cdot P(O_{ij}=0|x) \\ &= P(O_{ij}=1|x) \\ &+ (1-max_c(P(Y_{ij}=c|x))\cdot p(O_{ij}=0|x). \end{aligned} \tag{5.2}$$

We also assume that $P(\text{outlier}_{ij}|O_{ij}=1)=1$, that is to say that our outlier detection head does not have false positives. To correct the false negatives, we use max-softmax to get $P(\text{outlier}_{ij}|O_{ij}=0)$.

We can also treat the two-head model as an ensemble, in which case we calculate the outlier probability as a mean of outlier probabilities given by each head:

$$P(\text{outlier}_{ij}|x) = \frac{P(O_{ij}=1|x)+(1-max_c(P(Y_{ij}=c|x))}{2} \tag{5.3}$$

Finally, we can interpret the outlier predictions given by the outlier detection and the segmentation as independent and calculate the probability that both heads will predict that a pixel is an outlier as the product of the two probabilities:

$$P(\text{outlier}_{ij}|x) = P(O_{ij}=1|x)\cdot(1-max_c(P(Y_{ij}=c|x)) \tag{5.4}$$

The results of outlier detection using the $\text{LDN\_BIN}_{JS,RSP}$ when combining max-softmax with the output of outlier detection head can be seen in Table 5.15. The combined probability defined in Equation 5.4 performs best on all of the datasets. Max-softmax is able to detect small outliers, while the outlier detection head prevents false positives on semantic borders.

Table 5.15: Results of LDN_BIN$_{JS,RSP}$ on Lost&Found, StreetHazard and Vistas-NP when using only the output of the outlier detection head as well as when that output is combined with max-softmax using equations 5.2 (total probability), 5.3 (ensemble) and 5.4 (OP×MSM).

Subset	Criterion	AP	FPR95
FS LF	OP	36.9	20.0
	total probability	16.3	27.4
	ensemble	19.1	27.4
	OP×MSM	**39.7**	**16.4**
StreetHazard	OP	19.7	56.2
	total probability	12.5	27.2
	ensemble	12.9	**27.2**
	OP×MSM	**20.6**	46.9
Vistas-NP	OP	10.8	33.2
	total probability	15.5	17.2
	ensemble	18.0	17.2
	OP×MSM	**24.0**	**13.7**

5.6.8 Experiments on WildDash benchmark

Table 5.16 presents open-set recognition results on the WildDash benchmark. Our models are listed in the last three rows of the table. The LDN_OE model has a single C-way multi-class head and uses max-softmax for outlier detection. The LDN_BIN and LDN_BIN$_{JS}$ both have two heads, where *JS* denotes scale jittering during training. All three models have been trained on Vistas train, Cityscapes train, and WildDash val (inliers) and ImageNet-1k-bb (noisy negatives).

LDN_BIN and LDN_OE differ only in outlier detection protocol, with the rest of the training setup being identical. The two-head model performs better in most classic evaluation categories as well as in the negative category, but has a lower meta-average score. This is caused by a larger performance drop in most hazard categories.

Qualitative analysis of the two models is shown in Figures 5.13, 5.14 and 5.15

The first four rows in Figure 5.13 show images taken in normal conditions, while the last two rows show outlier images. The C-way model tends to classify small objects (cf. poles in image in row 3) as well as distant objects (cf. trucks in the distance in the image in the third row) as outliers. This model makes more false outlier detections in typical traffic scenes. Furthermore,

Table 5.16: Open-set segmentation results on the WildDash 1 benchmark. All our submissions are on Cityscapes train, Vistas train and WildDash 1 val (inliers) and ImageNet-1k-bb (negatives). The pasting procedure included only fixed-sized patches. LDN_BIN denotes the two-head model, while LDN_OE denotes the C-way multi-class model. Only our best submission was trained with jitter scaling (JS).

Model	Meta Avg	Classic				Negative
	mIoU cla	mIoU cla	iIoU cla	mIoU cat	iIoU cat	mIoU cla
DRN_MPC [121]	28.3	29.1	13.9	49.2	29.2	15.9
DeepLabv3+_CS [122]	30.6	34.2	24.6	49.0	38.6	15.7
MapillaryAI_ROB [123]	38.9	41.3	38.0	60.5	57.6	25.0
AHiSS_ROB [124]	39.0	41.0	32.2	53.9	39.3	43.6
MSeg [83]	43.0	42.2	31.0	59.5	51.9	51.8
MSeg_1080 [83]	**48.3**	**49.8**	**43.1**	63.3	56.0	**65.0**
LDN_BIN (ours)	41.8	43.8	37.3	58.6	53.3	54.3
LDN_OE (ours)	42.7	43.3	31.9	60.7	50.3	52.8
$\text{LDN\_BIN}_{\text{JS}}$(ours)	46.9	48.8	42.8	**63.6**	**59.3**	47.7

the two-head model performs better in negative images. Output of the C-way model on negative images contains small patches not classified as outliers.

Figure 5.14 illustrates the impact of WildDash hazards on the output of the submitted models.

The results show that the two-head model is more sensitive to overexposure and distortion hazards, though it succeeds when the hazard is not severe (rows 1-4).

Row 5 contains an example of occlusion. Both of the models are able to successfully segment the torso or the person behind the pole but they struggle with the lower part. Row 6 contains an example of the particles hazard (rain on the windshield), while row 7 contains an example of the variation hazard (with a tank truck being an atypical example of a truck).

Figure 5.13: Qualitative performance on the WildDash benchmark. Each triplet contains a test image (left), the output of the two-head model (center), and the output of the C-way multi-class model trained to predict uniform distribution in outliers (right). The two-head model does not produce false positives at semantic borders (rows 1-4). Both models correctly recognize outliers in negative images (rows 5-6).

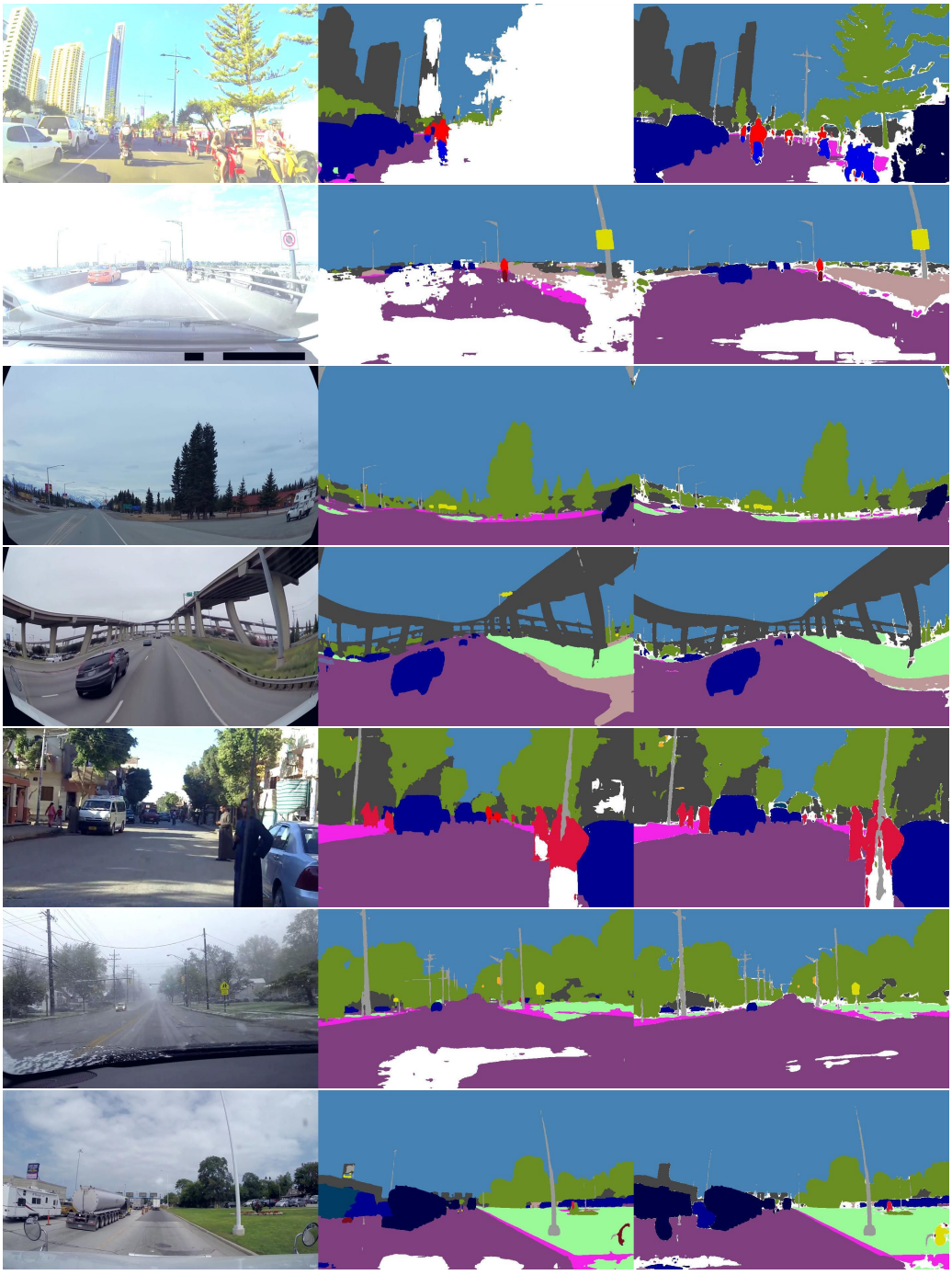

Figure 5.14: Qualitative performance on WildDash test images with overexposure and distortion hazards. Each triplet contains a test image (left), the output of the two-head model (center), and the output of the C-way multi-class model trained to predict uniform distribution in outliers (right). The two-head model rejects images with severe hazards.

Figure 5.15 shows failure modes. The first two rows show images of animals on roads. Animals are usually classified as pedestrians. The two-head model classifies most of the pixels of the image in row 1 as outliers. Both of the models fail to classify the animals as outliers accurately.

The windshield wiper in row 3 is classified as an inlier. This is because the submitted models were trained on Wilddash val, which contains examples of images with windshield wipers. Those pixels are ignored during training but they still influence the features extracted by the dense feature extractor. This view is supported by Figure 5.6, where images in column 3 demonstrate that a model trained only on Vistas classifies windshield wipers as outliers.

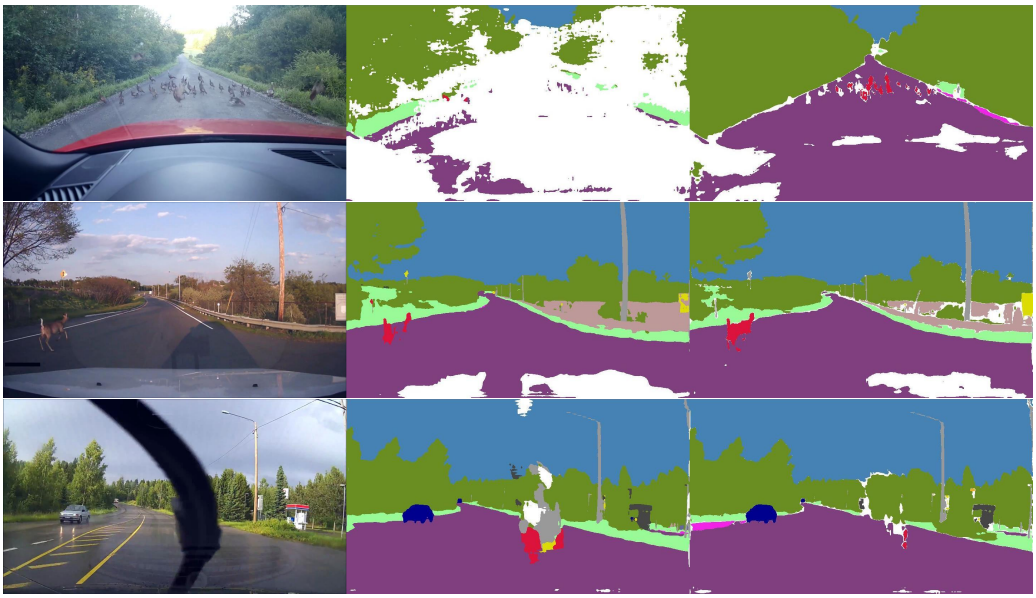

Figure 5.15: Qualitative performance of our two submissions to the WildDash benchmark. Each triplet contains a test image (left), the output of the two-head model (center), and the output of the model trained to predict uniform distribution in outliers (right). Rows 1-2 show that our current models are unable to correctly detect small outlier objects. Row 3 shows that the windshield wiper is recognized as inlier, which occurs due to its presence in WildDash val (cf. Figure 5.6).

The best overall performance is achieved by the MSeg_1080 submission. We note however that this model uses more negative supervision than us (densely labeled Ade20k and COCO vs bounding boxes in ImageNet-1k). Additionally, they train and perform inference on larger resolutions and use a model with almost 4 times more parameters (65.8M params compared to our 17.4M params). MSeg_1080 is also somewhat less sensitive to some hazards (most significantly underexposure) which may be due to their use of a significantly larger inlier training dataset: aside from Vistas and Cityscapes, they also use Berkeley DeepDrive [100] and Indian Driving Dataset [125]. Our model is competitive and actually outperforms MSeg in most of the categories when inference is done on similar resolution (MSeg vs LDN_BIN).

Figure 5.16 presents a detailed comparison between MSeg and LDN_BIN$_{JS}$ as shown on the WildDash benchmark. The columns show: i) original image, ii) MSeg output and iii)

LDN_BIN$_{JS}$ output. The second and third image show that Mseg performs better on segmentation of small objects in challenging conditions which is likely due to larger resolution. Note, however, that the MSeg model does not recognize the hood of the car (row 1) and black rectangles (row 3) as outliers. Detailed qualitative results for LDN_BIN and LDN_OE can be found in [98]. Note that LDN_BIN$_{JS}$ behaves similarly to LDN_OE in that it is able to find inlier objects in outlier context.

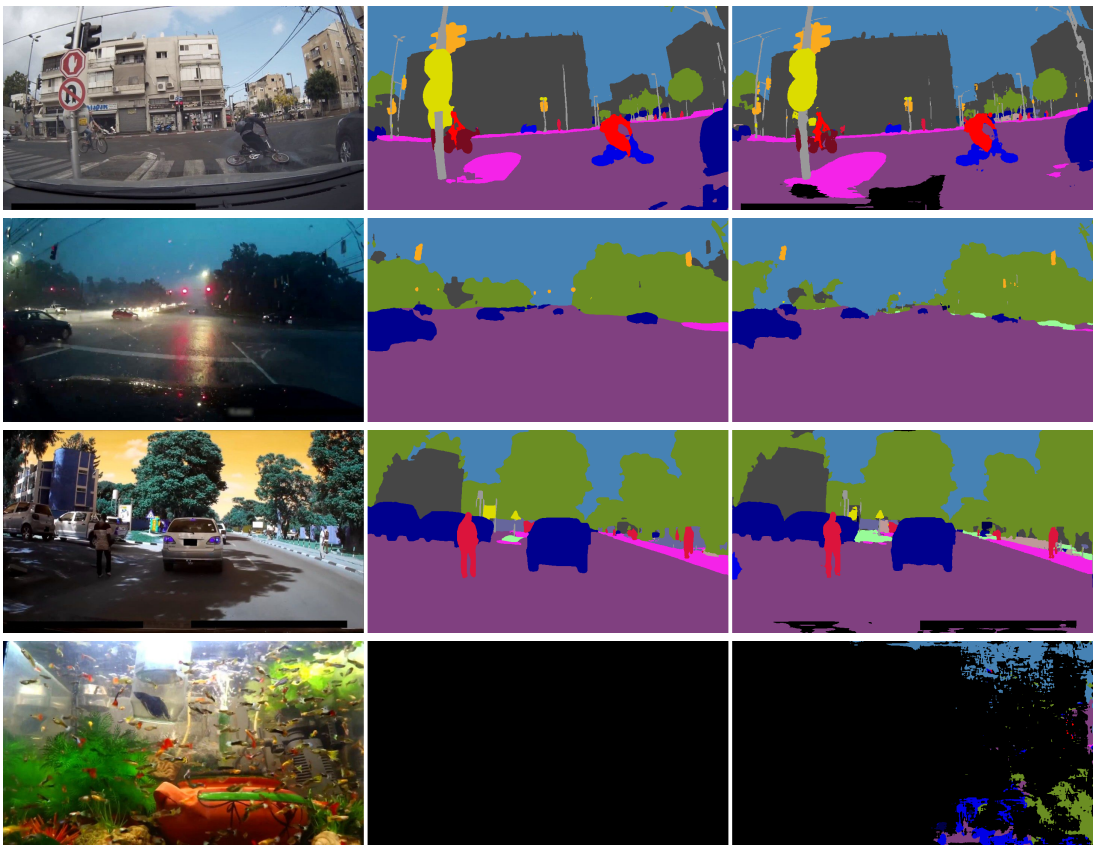

Figure 5.16: Qualitative comparison between MSeg (column 2) and LDN_BIN$_{JS}$(column 3) on the WildDash benchmark (column 1). MSeg performs better on natural negative images (line 4), and small objects (line 2), but it is unable to locate outlier patches in traffic images (lines 1 and 3).

5.6.9 Results on Fishyscapes benchmark

Table 5.17 shows current results on the Fishyscapes benchmark [13]. The benchmark provides mIoU score which is measured on Cityscapes validation dataset and AP and FPR95 on two datasets: Lost and Found and FS Static. Lost and Found comprises 300 images taken from the Lost and Found dataset, relabelled to distinguish between inlier, outlier and void classes and filtered to exclude sequences where road hazards are inlier classes (e.g. bicycles). FS static was created by pasting Pascal objects into Cityscapes images.

Notice that LDN_BIN$_{JS}$ is almost exactly the same model that was presented in Table 5.16. We needed to convert the model to Tensorflow 1 in order to comply with submission require-

ments. We lose a couple of points on our segmentation in this conversion due to a known issue with bilinear interpolation in Tensorflow 1.

LDN_BIN$_{\text{JS, RSP}}$ and LDN_BIN$_{\text{JS, RSP, ADE}}$ were both trained with randomly scaled pasted patches. The latter model, however, sampled negative data from ADE20k. The drop in FPR95 score indicates that data augmentations and appropriately chosen negative dataset improve outlier detection performance. We outperform other models by a large margin on FS static. We also achieve the second best outlier detection AP on Lost and Found images without the significant drop in segmentation performance that occurs in the best submission.

Table 5.17: Open-set segmentation results on the Fishyscapes benchmark. All our submissions are two-head models that were trained with scale jittering (JS) on Cityscapes train, Vistas train and WildDash 1 val (inliers) and ImageNet-1k-bb or ADE20K (negatives). Two of the models were trained with randomly sized pasted patches (RSP), and out od the two, one was trained with ADE20K negatives (ADE)

Model		Criterion	Train	OoD	City	Lost and Found		FS Static	
					mIoU	AP	FPR95	AP	FPR95
Dirichlet DeepLab [13]		prior entropy	✓	✓	70.5	**34.3**	47.4	31.3	84.6
Bayesian DeepLab [13]		mutual information	✓	✗	73.8	9.8	38.5	48.7	15.5
OoD training [13]		maximize entropy	✓	✓	79.0	1.7	30.6	27.5	23.6
Softmax [13]		entropy	✗	✗	80.0	2.9	44.8	15.4	39.8
		max-softmax (MSM)	✗	✗		1.8	44.9	12.9	39.8
Learned embedding density [13]		logistic regression	✗	✓	80.0	4.7	24.4	57.2	13.4
		minimum nll	✗	✗		4.3	47.2	62.1	17.4
		single-layer nll	✗	✗		3.0	32.9	40.9	21.3
Image resynthesis		resynthesis difference	✗	✗	**81.4**	5.7	48.1	29.6	27.1
Discriminative outlier detection head (ours)	LDN_BIN$_{\text{JS}}$	outlier probability (OP)	✓	✓	77.7	15.7	76.9	82.9	5.1
	LDN_BIN$_{\text{JS, RSP}}$	outlier probability (OP)	✓	✓	77.3	21.2	36.9	86.2	2.4
		OP $\times$ MSM	✓	✓		30.9	22.2	84.0	10.3
	LDN_BIN$_{\text{JS, RSP, ADE}}$	OP $\times$ MSM	✓	✓		31.3	**19.0**	**96.8**	**0.3**

5.7 Discussion

We have presented a model for open-set semantic segmentation. Our architecture builds an open-set recognition module on top of a dense feature extractor. We find that the two-head module which separates semantic segmentation and outlier detection performs the best since it offers the best combination of semantic segmentation and outlier detection accuracy (cf. Table 5.8). Furthermore, qualitative analysis indicates that it is the only approach that does not yield a

high outlier probability on semantic borders (cf. Figure 5.6). The outputs of the two heads may be combined for improved performance on small outliers (cf. 5.15).

We show that outlier detection may be realised as a simple binary classifier that distinguishes between inlier data and a general-purpose negative dataset such as ImageNet-1k. This approach works on a variety of datasets without modifications to the choice of negative data (cf. Tables 5.2, 5.6, 5.7, 5.13 and 5.14).

Standalone discriminative outlier detection is prone to overfitting and works best with some form of regularization, such as reduction in capacity (cf. Table 5.4). It thus benefits from sharing features with a segmentation task (cf. Table 5.2 vs Table 5.8).

Our experiments indicate that open-set segmentation performance is impacted by the type of negative data seen during training. Overall, negative data significantly improves outlier detection (cf. Table 5.1 vs. Table 5.8). What is more, for the model to have the ability to detect outlier patches in mixed-content images, such images must be used during training (cf. Table 5.2). We are able to create mixed content images by pasting negative data into inlier images. We get the best results when we vary the size of the pasted data since it enables the detection of smaller outlier patches (cf. Tables 5.12 and 5.13).

We do not curate the negative dataset in any way, and the noise may impact the model performance. Some of the noise may be reduced by ignoring areas of negative images where there is the most overlap between the inlier and outlier data (cf. Table 5.11). Still, that is not always possible. We therefore propose a batch formation that ensures that the training batches always contain an approximately equal share of inlier and negative pixels. Since the negative dataset is usually much larger and significantly more diverse than the inlier one, many inlier epochs are performed during one negative epoch. This means that occasional inlier pixels in negative images will be comparatively rarely erroneously labeled as outliers. Our results show that we were able to successfully train models without a significant in segmentation or outlier detection performance (cf. Table 5.17 and 5.16).

Chapter 6

Conclusion

We propose a novel discriminative approach for dense outlier detection and open-set recognition. The proposed approach discriminates between an application-specific inlier dataset (e.g., Vistas, Cityscapes) and a diverse general-purpose negative dataset (e.g., ImageNet-1k). Our approach treats pixels from the general-purpose dataset as noisy, test-agnostic negative samples and trains on mixed batches with an approximately equal share of inliers and noisy negatives. This promotes robustness to occasional inlier content in negative images and facilitates stable development of batch normalization statistics. We encourage correct recognition of spatial borders between outlier and inlier pixels by pasting negative patches at random locations in inlier images. The resulting models generalize well to test images with anomalies of arbitrary shape.

We have successfully implemented the proposed dense open-set recognition approach as a multi-task model that performs outlier detection and semantic segmentation on top of shared features. This implementation allows us to perform dense open-set recognition with a single forward pass, without deteriorating the performance of either task. The decoupling of the two tasks increases the robustness of the primary task to the noise in the negative dataset. We have submitted the results of our best model to FishyScapes and WildDash benchmarks, where it is still the only method that competes at both benchmarks. Our model is currently at the top of the Fishyscapes Static leaderboard and is a close runner-up on WildDash 1, even though it is trained with less supervision than the top-ranking algorithm [83]. We also report successful dense outlier detection performance on the UCSD anomaly detection dataset, as well as Vistas-NP and StreetHazard open-set segmentation datasets. Most of our reported experiments feature the same model, hyperparameters, training procedure, and negative dataset; the only difference is the inliers being used. The only exception is the UCSD anomaly dataset, where we had to decrease the model capacity due to fewer training data.

We present several validation and ablation experiments and offer additional insights into our approach. Specifically, we show how the choice of training data and model capacity affects model performance. To address the potential drawbacks of our approach, we analyze its gener-

alization potential by testing on outliers not seen in the negative datasets. We successfully detect out-of-distribution samples not seen in the training data such as carts in the UCSD dataset. Furthermore, we explore whether training on synthetic mixed content images introduces unwanted side-effects, such as relying on pasting cues rather than outlier patch semantics for recognition. Finally, we examine the impact of the size of anomalies to outlier detection performance and propose additional methods for improving the recognition of small anomalies. Overall, our experiments provide a comprehensive insight into our method and highlight its strengths and limitations.

The results reported in this study provide strong evidence for our hypotheses that i) the use of noisy negatives can significantly improve dense outlier detection and open-set recognition, and ii) the resulting open-set models perform comparably to their closed-set counterparts in terms of closed-set mIoU. Based on the experimental results, we developed a multi-head open-set recognition model based on shared features between outlier detection and semantic segmentation. Our simple technique for creating mixed-content images through pasting promotes learning of accurate detection of out-of-distribution objects. Finally, our batch creation procedure decreases the impact of semantic noise in negative learning examples.

We acknowledge that there are still challenges to be addressed, particularly in detecting small outliers and relaxing the dependence on real negative data. Moving forward, it would be valuable to explore new methods to address these challenges, as well as investigate the potential benefits of leveraging recent advances in computer vision architectures such as transformers and vision-language models for more versatile outlier detection and better generalization potential.

Bibliography

[1] Brostow, G. J., Shotton, J., Fauqueur, J., Cipolla, R., "Segmentation and recognition using structure from motion point clouds", in ECCV, 2008, pp. 44-57.

[2] Neuhold, G., Ollmann, T., Bulò, S. R., Kontschieder, P., "The mapillary vistas dataset for semantic understanding of street scenes", in ICCV, 2017, pp. 5000-5009.

[3] Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A., "Scene parsing through ade20k dataset", in CVPR, 2017, pp. 633–641.

[4] Zhang, P., Liu, W., Wang, H., Lei, Y., Lu, H., "Deep gated attention networks for large-scale street-level scene segmentation", Pattern Recognition, Vol. 88, 2019, pp. 702–714.

[5] Sanjeewani, T. G. P., Verma, B. K., "Learning and analysis of ausrap attributes from digital video recording for road safety", in IVCNZ, 2019, pp. 1–6.

[6] Aksoy, Y., Oh, T., Paris, S., Pollefeys, M., Matusik, W., "Semantic soft segmentation", ACM Trans. Graph., Vol. 37, No. 4, 2018, pp. 72:1–72:13.

[7] Ronneberger, O., Fischer, P., Brox, T., "U-net: Convolutional networks for biomedical image segmentation", in MICCAI, 2015, pp. 234–241.

[8] Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., Boult, T. E., "Toward open set recognition", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 35, No. 7, 2013, pp. 1757–1772.

[9] van Amersfoort, J., Smith, L., Jesson, A., Key, O., Gal, Y., "On feature collapse and deep kernel learning for single forward pass uncertainty", arXiv preprint arXiv:2102.11409, 2021.

[10] Lucas, T., Shmelkov, K., Karteek, A., Schmid, C., Verbeek, J., "Adaptive density estimation for generative models", in Neural Information Processing Systems, 2019.

[11] Huang, H., Wang, Y., Hu, Q., Cheng, M.-M., "Class-specific semantic reconstruction for open set recognition", IEEE Trans. Pattern Anal. Mach. Intell., Vol. PP, August 2022.

[12] Zendel, O., Honauer, K., Murschitz, M., Steininger, D., Fernandez Dominguez, G., "Wilddash - creating hazard-aware benchmarks", in ECCV, September 2018, pp. 407–421.

[13] Blum, H., Sarlin, P.-E., Nieto, J., Siegwart, R., Cadena, C., "Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving", in ICCVW, 2019, pp. 2403-2412.

[14] Chan, R., Lis, K., Uhlemeyer, S., Blum, H., Honari, S., Siegwart, R., Fua, P., Salzmann, M., Rottmann, M., "Segmentmeifyoucan: A benchmark for anomaly segmentation", in NeurIPS, Vol. 1, 2021.

[15] Hendrycks, D., Gimpel, K., "A baseline for detecting misclassified and out-of-distribution examples in neural networks", in ICLR, 2017.

[16] Kendall, A., Gal, Y., "What uncertainties do we need in bayesian deep learning for computer vision?", in NIPS, 2017, pp. 5574–5584.

[17] Bevandic, P., Kreso, I., Orsic, M., Segvic, S., "Discriminative out-of-distribution detection for semantic segmentation", arXiv preprint arXiv:1808.07703, 2018.

[18] Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., Steger, C., "he mvtec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection", International Journal of Computer Vision, Vol. 129, April 2021.

[19] Bishop, C. M., "Novelty detection and neural network validation", IEE Proceedings - Vision, Image and Signal Processing, Vol. 141, 1994, pp. 217-222.

[20] Dinh, L., Sohl-Dickstein, J., Bengio, S., "Density estimation using real NVP", in ICLR, 2017.

[21] Ho, J., Jain, A., Abbeel, P., "Denoising diffusion probabilistic models", in NeurIPS, 2020, pp. 6840–6851.

[22] Du, Y., Mordatch, I., "Implicit generation and modeling with energy based models", in NeurIPS, Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., (ur.), Vol. 32, 2019.

[23] Kingma, D. P., Welling, M., "Auto-encoding variational bayes", in ICLR, 2014.

[24] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., Bengio, Y., "Generative adversarial nets", in NeurIPS, 2014, pp. 2672–2680.

[25] Ruff, L., Kauffmann, J., Vandermeulen, R., Montavon, G., Samek, W., Kloft, M., Dietterich, T., Müller, K.-R., "A unifying review of deep and shallow anomaly detection", Proceedings of the IEEE, Vol. PP, 02 2021, pp. 1-40.

[26] Nalisnick, E. T., Matsukawa, A., Teh, Y. W., Görür, D., Lakshminarayanan, B., "Do deep generative models know what they don't know?", in ICLR, 2019.

[27] Kirichenko, P., Izmailov, P., Wilson, A. G., "Why normalizing flows fail to detect out-of-distribution data", in NeurIPS, 2020, pp. 20 578–20 589.

[28] Lis, K., Nakka, K., Fua, P., Salzmann, M., "Detecting the unexpected via image resynthesis", in ICCV, 2019, pp. 2152-2161.

[29] Di Biase, G., Blum, H., Siegwart, R., Cadena, C., "Pixel-wise anomaly detection in complex driving scenes", in CVPR, 2021, pp. 16 913-16 922.

[30] Vojir, T., Šipka, T., Aljundi, R., Chumerin, N., Reino, D. O., Matas, J., "Road anomaly detection by partial image reconstruction with segmentation coupling", in 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 15 631-15 640.

[31] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Fei-Fei, L., "ImageNet Large Scale Visual Recognition Challenge", Inter. Jour. of Comput. Vision, Vol. 115, No. 3, 2015, pp. 211–252.

[32] Rosenblatt, F., "The perceptron: A probabilistic model for information storage and organization in the brain", Psychological Review, 1958, pp. 65–386.

[33] He, K., Zhang, X., Ren, S., Sun, J., "Spatial pyramid pooling in deep convolutional networks for visual recognition", in ECCV, 2014, pp. 346-361.

[34] Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., "Pyramid scene parsing network", in CVPR, 2017, pp. 6230-6239.

[35] He, K., Zhang, X., Ren, S., Sun, J., "Deep residual learning for image recognition", in CVPR, 2016, pp. 770-778.

[36] Huang, G., Liu, Z., Weinberger, K. Q., "Densely connected convolutional networks", in CVPR, 2017, pp. 2261-2269.

[37] Krešo, I., Krapac, J., Segvic, S., "Efficient ladder-style densenets for semantic segmentation of large images", IEEE Trans. Intell. Transp. Syst., 2020, pp. 1–11.

[38] Hawkins, D. M., Identification of Outliers. Springer, 1980.

[39] Pimentel, M., Clifton, D., Clifton, L., Tarassenko, L., "A review of novelty detection", Signal Processing, Vol. 99, 2014, pp. 215–249.

[40] Ren, J., Liu, P., Fertig, E., Snoek, J., Poplin, R., DePristo, M., Dillon, J., Lakshminarayanan, B., "Likelihood ratios for out-of-distribution detection", in NeurIPS, 2019, pp. 14 707–14 718.

[41] Grathwohl, W., Wang, K., Jacobsen, J., Duvenaud, D., Norouzi, M., Swersky, K., "Your classifier is secretly an energy based model and you should treat it like one", in ICLR, 2020.

[42] Serrà, J., Álvarez, D., Gómez, V., Slizovskaia, O., Núñez, J. F., Luque, J., "Input complexity and out-of-distribution detection with likelihood-based generative models", in ICLR, 2020.

[43] Zhang, L. H., Goldstein, M., Ranganath, R., "Understanding failures in out-of-distribution detection with deep generative models", Proceedings of machine learning research, Vol. 139, 2021, pp. 12 427-12 436.

[44] Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., Langs, G., "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery", in IPMI, 2017, pp. 146–157.

[45] Zenati, H., Romain, M., Foo, C., Lecouat, B., Chandrasekhar, V., "Adversarially learned anomaly detection", in ICDM, 2018, pp. 727-736.

[46] Hinton, G. E., "Connectionist learning procedures", Artificial Intelligence, Vol. 40, No. 1, 1989, pp. 185-234.

[47] Japkowicz, N., Myers, C., Gluck, M., "A novelty detection approach to classification", Proceedings of the Fourteenth Joint Conference on Artificial Intelligence, 10 1999.

[48] Sabokrou, M., Khalooei, M., Fathy, M., Adeli, E., "Adversarially learned one-class classifier for novelty detection", in CVPR, 2018, pp. 3379–3388.

[49] Akcay, S., Atapour-Abarghouei, A., Breckon, T. P., "Ganomaly: Semi-supervised anomaly detection via adversarial training", in ACCV, 2019, pp. 622–637.

[50] Baur, C., Wiestler, B., Albarqouni, S., Navab, N., "Deep autoencoding models for unsupervised anomaly segmentation in brain mr images", in MICAAI BrainLes, 2019, pp. 161–169.

[51] Soukup, D., Pinetz, T., “Reliably decoding autoencoders’ latent spaces for one-class learning image inspection scenarios”, in Proceedings of the OAGM Workshop 2018, Medical Image Analysis, 05 2018.

[52] Ionescu, R. T., Khan, F. S., Georgescu, M.-I., Shao, L., “Object-centric auto-encoders and dummy anomalies for abnormal event detection in video”, in CVPR, June 2019, pp. 7834-7843.

[53] Nguyen, T. N., Meunier, J., “Anomaly detection in video sequence with appearance-motion correspondence”, in ICCV, 2019, pp. 1273-1283.

[54] Park, H., Noh, J., Ham, B., “Learning memory-guided normality for anomaly detection”, in CVPR, June 2020, pp. 14 360-14 369.

[55] Liu, W., Luo, W., Lian, D., Gao, S., “Future frame prediction for anomaly detection - a new baseline”, CVPR, 2018, pp. 6536-6545.

[56] Andrews, J. T. A., Tanay, T., Morton, E. J., Griffin, L. D., “Transfer representation-learning for anomaly detection”, in ICML, 2016.

[57] Nazaré, T. S., de Mello, R. F., Ponti, M. A., “Are pre-trained cnns good feature extractors for anomaly detection in surveillance videos?”, arXiv preprint arXiv:1811.08495, 2018.

[58] Bergmann, P., Fauser, M., Sattlegger, D., Steger, C., “Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings”, in CVPR, 2020, pp. 4182-4191.

[59] Li, Z., Hoiem, D., “Learning without forgetting”, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 40, No. 12, 2018, pp. 2935–2947.

[60] Perera, P., Patel, V. M., “Learning deep features for one-class classification”, IEEE Transactions on Image Processing, Vol. 28, No. 11, 2019, pp. 5450-5463.

[61] Zhang, H., Li, A., Guo, J., Guo, Y., “Hybrid models for open set recognition”, in ECCV, 2020.

[62] Bendale, A., Boult, T. E., “Towards open set deep networks”, in CVPR, 2016, pp. 1563-1572.

[63] Lee, K., Lee, K., Lee, H., Shin, J., “A simple unified framework for detecting out-of-distribution samples and adversarial attacks”, in NeurIPS, 2018, pp. 7167–7177.

[64] Júnior, P. R. M., de Souza, R. M., de Oliveira Werneck, R., Stein, B. V., Pazinato, D. V., de Almeida, W. R., Penatti, O. A. B., da Silva Torres, R., Rocha, A., "Nearest neighbors distance ratio open-set classifier", Machine Learning, Vol. 106, 2016, pp. 359-386.

[65] Matan, O., Kiang, R., Stenard, C. E., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., Jackel, L. D., LeCun, Y., "Handwritten character recognition using neural network architectures", in USPSATC, 1990, pp. 1003-1011.

[66] Guo, C., Pleiss, G., Sun, Y., Weinberger, K. Q., "On calibration of modern neural networks", in ICML, 2017, pp. 1321–1330.

[67] Liang, S., Li, Y., Srikant, R., "Enhancing the reliability of out-of-distribution image detection in neural networks", in ICLR, 2018.

[68] Hsu, Y.-C., Shen, Y., Jin, H., Kira, Z., "Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data", in CVPR, 2020, pp. 10 948-10 957.

[69] Gal, Y., Ghahramani, Z., "Dropout as a bayesian approximation: Representing model uncertainty in deep learning", in ICML, 2016, pp. 1050–1059.

[70] Alex Kendall, V. B., Cipolla, R., "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding", in BMVC, September 2017, pp. 57.1-57.12.

[71] Lakshminarayanan, B., Pritzel, A., Blundell, C., "Simple and scalable predictive uncertainty estimation using deep ensembles", in NeurIPS, 2017, pp. 6402–6413.

[72] Vyas, A., Jammalamadaka, N., Zhu, X., Das, D., Kaul, B., Willke, T. L., "Out-of-distribution detection using an ensemble of self supervised leave-out classifiers", in ECCV, 2018, pp. 560–574.

[73] DeVries, T., Taylor, G. W., "Learning confidence for out-of-distribution detection in neural networks", arXiv preprint arXiv:1802.04865, 2018.

[74] Smith, L., Gal, Y., "Understanding measures of uncertainty for adversarial example detection", in UAI, 2018, pp. 560-569.

[75] Malinin, A., Gales, M., "Predictive uncertainty estimation via prior networks", in NeurIPS, 2018, pp. 7047–7058.

[76] Lee, K., Lee, H., Lee, K., Shin, J., "Training confidence-calibrated classifiers for detecting out-of-distribution samples", in ICLR, 2018.

[77] Hendrycks, D., Mazeika, M., Dietterich, T., “Deep anomaly detection with outlier exposure”, in ICLR, 2019.

[78] Torralba, A., Efros, A. A., “Unbiased look at dataset bias”, in CVPR, 2011, pp. 1521-1528.

[79] Neal, L., Olson, M., Fern, X., Wong, W.-K., Li, F., “Open set learning with counterfactual images”, in ECCV, 2018, pp. 613-628.

[80] Ge, Z., Demyanov, S., Chen, Z., Garnavi, R., “Generative openmax for multi-class open set classification”, arXiv preprint arXiv:1707.07418, 2017.

[81] Chan, R., Rottmann, M., Gottschalk, H., “Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation”, 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 5108-5117.

[82] Zhang, X., LeCun, Y., “Universum prescription: Regularization using unlabeled data”, AAAI Conference on Artificial Intelligence, Vol. 31, 11 2015.

[83] Lambert, J., Zhuang, L., Sener, O., Hays, J., Koltun, V., “MSeg: A composite dataset for multi-domain semantic segmentation”, in CVPR, 2020, pp. 2876-2885.

[84] Kreso, I., Orsic, M., Bevandic, P., Segvic, S., “Robust semantic segmentation with ladder-densenet models”, arXiv preprint arXiv:1806.03465, 2018.

[85] Vaze, S., Han, K., Vedaldi, A., Zisserman, A., “Open-set recognition: A good closed-set classifier is all you need”, in International Conference on Learning Representations, 2022, dostupno na: https://openreview.net/forum?id=5hLP5JY9S2d

[86] Bevandić, P., Oršić, M., Grubišić, I., Šarić, J., Šegvić, S., “Multi-domain semantic segmentation with overlapping labels”, in WACV, January 2022, pp. 2615-2624.

[87] Brock, A., Donahue, J., Simonyan, K., “Large scale GAN training for high fidelity natural image synthesis”, in ICLR, 2019.

[88] Grcic, M., Bevandic, P., Segvic, S., “Dense open-set recognition with synthetic outliers generated by real nvp”, in VISIGRAPP (4: VISAPP), 2021, pp. 133-143, dostupno na: https://doi.org/10.5220/0010260701330143

[89] Shafaei, A., Schmidt, M., Little, J. J., “Does your model know the digit 6 is not a cat? A less biased evaluation of "outlier" detectors”, CoRR, Vol. abs/1809.04729, 2018.

[90] Caruana, R., “Multitask learning”, Machine Learning, Vol. 28, No. 1, Jul 1997, pp. 41–75.

[91] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Y. Ng, A., "Multimodal deep learning", in ICML, 2011, pp. 689-696.

[92] Bengio, Y., Courville, A. C., Vincent, P., "Representation learning: A review and new perspectives", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 35, No. 8, 2013, pp. 1798–1828.

[93] Zamir, A. R., Sax, A., Shen, W. B., Guibas, L. J., Malik, J., Savarese, S., "Taskonomy: Disentangling task transfer learning", in CVPR, 2018.

[94] Eigen, D., Fergus, R., "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture", ICCV, 2015, pp. 2650-2658.

[95] Dvornik, N., Shmelkov, K., Mairal, J., Schmid, C., "BlitzNet: A real-time deep network for scene understanding", in ICCV, 2017.

[96] He, K., Gkioxari, G., Dollár, P., Girshick, R., "Mask R-CNN", in ICCV, 2017.

[97] Xia, Y., Zhang, Y., Liu, F., Shen, W., Yuille, A., "Synthesize then compare: Detecting failures and anomalies for semantic segmentation", in ECCV, 2020.

[98] Bevandic, P., Kreso, I., Orsic, M., Segvic, S., "Simultaneous semantic segmentation and outlier detection in presence of domain shift", in GCPR, 2019, pp. 33–47.

[99] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., "The cityscapes dataset for semantic urban scene understanding", in CVPR, 2016, pp. 3213–3223.

[100] Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., Darrell, T., "BDD100K: A diverse driving video database with scalable annotation tooling", arXiv preprint arXiv:1805.04687, 2018.

[101] Bevandić, P., Krešo, I., Oršić, M., Šegvić, S., "Dense open-set recognition based on training with noisy negative images", Image and Vision Computing, Vol. 124, 2022, pp. 104490, dostupno na: https://www.sciencedirect.com/science/article/pii/S0262885622001196

[102] Everingham, M., Eslami, S. M. A., Gool, L. V., Williams, C. K. I., Winn, J. M., Zisserman, A., "The pascal visual object classes challenge: A retrospective", International Journal of Computer Vision, Vol. 111, No. 1, 2015, pp. 98–136.

[103] Hendrycks, D., Basart, S., Mazeika, M., Mostajabi, M., Steinhardt, J., Song, D., "A benchmark for anomaly segmentation", arXiv preprint arXiv:1911.11132, 2019.

[104] Angus, M., Czarnecki, K., Salay, R., "Efficacy of pixel-level OOD detection for semantic segmentation", arXiv preprint arXiv:1911.02897, 2019.

[105] Pinggera, P., Ramos, S., Gehrig, S., Franke, U., Rother, C., Mester, R., "Lost and found: detecting small road hazards for self-driving vehicles", in IROS, 2016, pp. 1099 - 1106.

[106] Grcić, M., Bevandić, P., Šegvić, S., "Densehybrid: Hybrid anomaly detection for dense open-set recognition", in ECCV, 2022.

[107] Badrinarayanan, V., Kendall, A., Cipolla, R., "Segnet: A deep convolutional encoder-decoder architecture for image segmentation", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 39, No. 12, 2017, pp. 2481–2495.

[108] Kreso, I., Krapac, J., Segvic, S., "Ladder-style densenets for semantic segmentation of large natural images", in ICCV CVRSUAD, 2017, pp. 238–245.

[109] Lin, T., Dollár, P., Girshick, R. B., He, K., Hariharan, B., Belongie, S. J., "Feature pyramid networks for object detection", in CVPR, 2017, pp. 936–944.

[110] Franchi, G., Bursuc, A., Aldea, E., Dubuisson, S., Bloch, I., "One versus all for deep neural network incertitude (OVNNI) quantification", aarXiv:2006.00954, 2020.

[111] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., "ImageNet: A Large-Scale Hierarchical Image Database", in CVPR09, 2009.

[112] Everingham, M., Gool, L., Williams, C. K., Winn, J., Zisserman, A., "The pascal visual object classes (voc) challenge", Int. J. Comp. Vis., Vol. 88, 2010, pp. 303–338.

[113] Yu, F., Zhang, Y., Song, S., Seff, A., Xiao, J., "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop", arXiv preprint arXiv:1506.03365, 2015.

[114] Hendrycks, D., Basart, S., Mazeika, M., Zou, A., Kwon, J., Mostajabi, M., Steinhardt, J., Song, D., "Scaling out-of-distribution detection for real-world settings", in ICML, Vol. 162, 2022, pp. 8759–8773.

[115] Sugiyama, M., Borgwardt, K., "Rapid distance-based outlier detection via sampling", in Advances in Neural Information Processing Systems 26, Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. Q., (ur.). Curran Associates, Inc., 2013, pp. 467–475, dostupno na: http://papers.nips.cc/paper/5127-rapid-distance-based-outlier-detection-via-sampling.pdf

[116] Ionescu, R. T., Smeureanu, S., Alexe, B., Popescu, M., "Unmasking the abnormal events in video", 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2914-2922.

[117] Pang, G., Yan, C., Shen, C., Hengel, A. v. d., Bai, X., "Self-trained deep ordinal regression for end-to-end video anomaly detection", in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[118] Liu, Y., Li, C.-L., Póczos, B., "Classifier two sample test for video anomaly detections", in BMVC, 2018.

[119] Franchi, G., Bursuc, A., Aldea, E., Dubuisson, S., Bloch, I., "Tradi: Tracking deep neural network weight distributions", in ECCV, 2020, pp. 105-121.

[120] Zhu, R., Zhang, S., Wang, X., Wen, L., Shi, H., Bo, L., Mei, T., "Scratchdet: Training single-shot object detectors from scratch", in CVPR, 2019, pp. 2263-2272.

[121] Yu, F., Koltun, V., Funkhouser, T., "Dilated residual networks", in CVPR, 2017, pp. 636-644.

[122] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., "Encoder-decoder with atrous separable convolution for semantic image segmentation", in ECCV, 2018, pp. 801-818.

[123] Bulò, S. R., Porzi, L., Kontschieder, P., "In-place activated batchnorm for memory-optimized training of dnns", in CVPR, 2017, pp. 5639-5647.

[124] Meletis, P., Dubbelman, G., "Training of convolutional networks on multiple heterogeneous datasets for street scene semantic segmentation", in IV, 2018, pp. 1045-1050.

[125] Varma, G., Subramanian, A., Namboodiri, A. M., Chandraker, M., Jawahar, C. V., "IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments", in WACV, 2019, pp. 1743–1751.

Biography

Petra Bevandić, a native of Zagreb, was born in 1990. She earned her MSc degree from the University of Zagreb in 2014 and began working as a Software Developer at Ericsson Nikola Tesla thereafter. In 2017, she returned to the Faculty of Electrical Engineering and Computing where she presently serves as a teaching assistant, dividing her time between research and coursework.

She has participated in two research projects, MULTICLOD and DATACROSS. Moreover, she has been a member of teams that have successfully competed in the Robust Vision Challenge Competition. These experiences have played an integral role in her ongoing work concerning open-set recognition and multi-dataset training. She also dedicates her time as a reviewer for international conferences and scientific journals.

Her research interests are focused on the development of robust and interpretable dense recognition models, multi-dataset and multi-modal training, and deep model transferability between different tasks and domains.

List of publications

Journal papers

1. **Bevandić, P., Krešo, I., Oršić, M., Šegvić, S., "Dense outlier detection and open-set recognition based on training with noisy negative images", Image and Vision Computing, Vol 124, 2022, p. 104490**
2. Sikirić, I., Brkić, K., Bevandić, P., Krešo, I., Krapac, J., Šegvić, S., "Traffic scene classification on a representation budget", IEEE Transactions on Intelligent Transportation Systems, 2019

Conference papers

1. **Bevandić, P., Krešo, I., Oršić, M., Šegvić, S., "Simultaneous semantic segmentation and outlier detection in presence of domain shift", in German Conference on Pattern Recognition. Springer, 2019, pp. 33–47**

2. Bevandić, P., Oršić, M., Grubišić, I., Šarić, J., Šegvić, S., "Multi-domain semantic segmentation with overlapping labels", in IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 2615-2624
3. Bevandić, P., Šegvić, S., "Automatic universal taxonomies for multi-domain semantic segmentation", in British Machine Vision Conference, 2022, pp. 2615-2624
4. Grcić, M., Bevandić, P., Šegvić, S., "Densehybrid: Hybrid anomaly detection for dense open-set recognition", in European Conference on Computer Vision, 2022, pp. 500-517
5. Grcić, M., Bevandić, P., Šegvić, S., "Dense open-set recognition with synthetic outliers generated by Real NVP", in International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2021, pp.122 - 143
6. Oršić, M., Krešo, I., Bevandić, P.,Šegvić, S., "In defense of pre-trained ImageNet architectures for real-time semantic segmentation of road-driving images", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, pp. 12 607–12 616

Other manuscripts

1. Bevandić, P., Krešo, I., Oršić, M., Šegvić, S., "Discriminative out-of-distribution detection for semantic segmentation", arXiv preprint arXiv:1808.07703, 2018
2. Bevandić, P., Oršić, M., Grubišić, I., Šarić, J., Šegvić, S., "Weakly supervised training of universal visual concepts for multi-domain semantic segmentation", arXiv preprint arXiv:2212.10340, 2022
3. Grcić, M., Bevandić, P., Kalafatić, Z., Šegvić, S., "Dense anomaly detection by robust learning on synthetic negative data", arXiv preprint arXiv:2011.12833, 2020
4. Krešo, I., Oršić, M., Bevandić, P., Šegvić, S., "Robust semantic segmentation with ladder-densenet models", arXiv preprint arXiv:1806.03465, 2018

Životopis

Petra Bevandić rođena je u Zagrebu 1990. godine. 2014. godine diplomirala je na Sveučilištu u Zagrebu. Nakon diplome, zapošljava se kao programer u kompaniji Ericsson Nikola Tesla. 2017. vraća se na Fakultet elektrotehnike i računarstva gdje radi kao asistent u nastavi balansirajući istraživački i nastavnički posao.

Sudjelovala je na dva istraživačka projekta: MULTICLOD i DATACROSS. Uz to, bila je dio ekipa koje su u nekoliko navrata sudjelovale na međunarodnom natjecanju Robust Vision Challenge Competition. Njezino istraživanje o predikciji nad otvorenim skupom podataka te učenju na više skupova podataka djelomično je inspirirano saznanjima koja su proizašla iz tih suradnji. Volontira kao recenzent za međunarodne konferencije i časopise.

Njezini istraživački interesi uključuju robusnost i interpretabilnost dubokih modela, treniranje na više skupova podataka, istovremeno treniranje više različitih zadataka te prijenos modela između zadataka i domena.