National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

Domain Generalization in Robust Vision Transformers for Semantic Segmentation in Autonomous Driving

Diploma Thesis

of

TZOKAS GEORGIOS

Supervisor: Athanasios Voulodimos
Assistant Professor, ECE NTUA

Athens, October 2024

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

Domain Generalization in Robust Vision Transformers for Semantic Segmentation in Autonomous Driving

Diploma Thesis

of

TZOKAS GEORGIOS

Supervisor: Athanasios Voulodimos
Assistant Professor, ECE NTUA

Approved by the examination committee on 10th October 2024.

(Signature)	*(Signature)*	*(Signature)*
.....	.....	.....
Athanasios Voulodimos	Georgios Stamou	Andreas-Georgios Stafylopatis
Assistant Professor, ECE NTUA	Professor, ECE NTUA	Professor, ECE NTUA

Athens, October 2024

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

(Signature)

.....

Tzokas Georgios

Διπλωματούχος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών ΕΜΠ

25th October 2024

Περίληψη

Η ταχεία πρόοδος της τεχνητής νοημοσύνης έχει οδηγήσει στην ευρεία υιοθέτηση της βαθιάς μάθησης σε σύγχρονες εφαρμογές. Η όραση υπολογιστών έχει επωφεληθεί ιδιαίτερα με εξαιρετικά αποτελεσματικά μοντέλα να χρησιμοποιούνται πλέον σε εφαρμογές πραγματικού χρόνου. Μία από αυτές τις εφαρμογές είναι η σημασιολογική κατάτμηση για την αυτόνομη οδήγηση, η οποία επιτρέπει στα αυτόνομα οχήματα να αποκτήσουν λεπτομερή κατανόηση του περιβάλλοντός, επιτρέποντάς τα να λαμβάνουν τεκμηριωμένες αποφάσεις σε πραγματικό χρόνο. Για τέτοιες εφαρμογές, είναι ζωτικής σημασίας τα μοντέλα να διατηρούν υψηλό επίπεδο ακρίβειας σε ποικίλες περιβαλλοντικές συνθήκες και να λειτουργούν σε πραγματικό χρόνο. Για να αμβλύνουμε αυτά τα προβλήματα χρησιμοποιούμε συνθετικά δεδομένα, ώστε να ξεπεράσουμε το εμπόδιο της συλλογής δεδομένων και του σχολιασμού. Η χρήση αγωγών γενίκευσης τομέων και ισχυρών μετασχηματιστών ενισχύει την ακρίβεια των μοντέλων σε διάφορα περιβάλλοντα. Τέλος, χρησιμοποιούμε έναν εξαιρετικά αποδοτικό αποκωδικοποιητή συνέλιξης για να ενισχύσουμε την ταχύτητα εξαγωγής συμπερασμάτων του μοντέλου, σε σύγκριση με άλλες αρχιτεκτονικές γενίκευσης. Με τη διεξαγωγή ενός πειράματος γενίκευσης παρουσιάζουμε ότι οι βελτιωμένες δυνατότητες σε πραγματικό χρόνο δεν έρχονται χωρίς θυσία στην ακρίβεια και τονίζουμε την ανάγκη περαιτέρω μείωσης της υπολογιστικής πολυπλοκότητας των μοντέλων μετασχηματιστών, ώστε να καταστούν βιώσιμη λύση για τη σημασιολογική κατάτμηση σε πραγματικό χρόνο στην αυτόνομη οδήγηση.

Λέξεις κλειδιά

Νευρωνικά Δίκτυα, Βαθιά Μάθηση, Κατάτμηση Εικόνας, Γενίκευση Πεδίου, Αυτόνομα Οχήματα, Σημασιολογική Τμηματοποίηση Πραγματικού Χρόνου

Abstract

Rapid advances in artificial intelligence have led to the widespread adoption of deep learning in modern applications. Computer vision has particularly benefited from highly efficient models that are now being used in real-time applications. One such application is semantic segmentation for autonomous driving, which allows autonomous vehicles to gain a detailed understanding of their environment, enabling them to make informed decisions in real time. For such applications, it is vital that models maintain a high level of accuracy in a variety of environmental conditions and operate in real-time. However, there are several problems that prevent the creation of models that meet the above conditions. These problems relate to the number and variety of training data, the complexity of real-world conditions, and the computational requirements of the architectures being exploited. To mitigate these problems we use synthetic data, to overcome the barrier of data-collection and annotation. The use of domain generalization pipelines and robust transformers enhances the models' accuracy across environments. Finally, we use a highly efficient convolutional decoder to enhance the model's inference speed, when compared to other generalization architectures. By conducting a generalization experiment we showcase that the improved real time capabilities come with no sacrifice to accuracy and emphasize the need to further reduce the computational complexity of transformer models, to make them a viable solution for real time semantic segmentation in autonomous driving.

Keywords

Neural Networks, Deep Learning, Image Segmentation, Field Generalization, Autonomous Vehicles, Real-Time Semantic Segmentation

to my parents

Acknowledgements

I would first like to thank Professor Athanasios Voulodimos, Professor Giorgos Stamou, and Professor Paraskevi Tzouveli for supervising this thesis and for giving me the opportunity to carry it out in the Artificial Intelligence and Learning Systems Laboratory. I also extend my special thanks to Nikolaos Spanos and Paraskevi Theofilou for their guidance and the excellent collaboration we had. I would like to thank my parents for their guidance and moral support throughout all these years. Finally, I would like to thank my friends, as I would not be where I am today without their support.

Athens, October 2024

Tzokas Georgios

Table of Contents

List of Images

List of Tables

Chapter 1

Εκτεταμένη Περίληψη στα Ελληνικά

Τα αυτόνομα οχήματα, εξοπλισμένα με κάμερες υψηλής ανάλυσης, μπορούν να αξιοποιήσουν μοντέλα μηχανικής μάθησης για λεπτομερή σημασιολογική τμηματοποίηση, επιτρέποντάς τους να εντοπίζουν και να κατηγοριοποιούν με ακρίβεια αντικείμενα και στοιχεία του δρόμου. Αυτά τα μοντέλα είναι ικανά να συλλαμβάνουν πολύπλοκες χωρικές σχέσεις, καθιστώντας τα πολύτιμα εργαλεία για την κατανόηση και αντίληψη του περιβάλλοντος. ιαχωρίζοντας οχήματα, πεζούς, σήματα κυκλοφορίας, λωρίδες και άλλα κρίσιμα στοιχεία του δρόμου, τα μοντέλα ελτιώνουν την ικανότητα του οχήματος να πλοηγείται με ασφάλεια, να σχεδιάζει έλτιστες διαδρομές και να λαμβάνει αποφάσεις σε πραγματικό χρόνο. Αυτό συνεισφέρει στην ακρίβεια και ασφάλεια των συστημάτων αυτόνομης οδήγησης. Παρ' όλα αυτά, τόσο στην απόκτηση των δεδομένων εκπαίδευσης όσο και στην εκπαίδευση αυτών των μοντέλων παρουσιάζονται σημαντικές προκλήσεις.

Τα μοντέλα μηχανικής μάθησης που προορίζονται για αυτόνομα οχήματα πρέπει να είναι εξίσου αποδοτικά και ακριβή. Πιο συγκεκριμένα, χρειάζεται να μπορούν να τμηματοποιούν δεκάδες καρέ ανά δευτερόλεπτο, ώστε το όχημα να λαμβάνει συνεχώς νέα δεδομένα για το περιβάλλον του. Επιπλέον, πρέπει να είναι σχεδιασμένα για να λειτουργούν σε ενσωματωμένα συστήματα, κάτι που σημαίνει ότι πρέπει να είναι σχετικά μικρά σε μέγεθος. Ταυτόχρονα, όμως, πρέπει να διατηρούν υψηλή ακρίβεια ώστε το όχημα να αντιλαμβάνεται με σαφήνεια τι συμβαίνει στο δρόμο. Η εκπαίδευση αυτών των μοντέλων απαιτεί μεγάλα σύνολα δεδομένων. Ωστόσο, τόσο η συλλογή όσο και η επισήμανση αυτών των δεδομένων αποτελούν μια ιδιαίτερα απαιτητική διαδικασία από πλευράς πόρων.

Για την επίτευξη μοντέλων που ανταποκρίνονται στα παραπάνω, έχουν προταθεί αρκετές τεχνικές και αρχιτεκτονικές με στόχο τη μείωση της πολυπλοκότητας των μοντέλων και του αριθμού των παραμέτρων τους. Αυτές οι τεχνικές επιτρέπουν τη δημιουργία αποδοτικότερων μοντέλων χωρίς να υσιάζεται η ακρίβεια. Για την αντιμετώπιση της έλλειψης δεδομένων, χρησιμοποιείται ασθενώς εποπτευόμενη μάθηση, όπου τα δεδομένα δεν είναι πλήρως επισημειωμένα, μειώνοντας έτσι τις απαιτήσεις σε πόρους. Μια άλλη προσέγγιση είναι η χρήση εικονικών δεδομένων κατά την εκπαίδευση, καθώς αυτά τα δεδομένα είναι εύκολο να δημιουγηθούν και να επισημανθούν. Ωστόσο, η χρήση εικονικών δεδομένων δημιουργεί μια νέα πρόκληση. Τα μοντέλα υποθέτουν ότι τα δεδομένα εκπαίδευσης και τα δεδομένα εφαρμογής προέρχονται από την ίδια κατανομή και είναι ευάλωτα στις ¨μετατοπίσεις πεδίου¨, δηλαδή στις αλλαγές μεταξύ των κατανομών.

Οι ¨μετατοπίσεις πεδίου¨ αποτελούν σοβαρό εμπόδιο κατά την εκπαίδευση και τον σχε-

διασμό ανθεκτικών μοντέλων. Ακόμα και σε περιπτώσεις, όπου τα δεδομένα εφαρμογής και εκπαίδευσης διαφέρουν έστω και σε αθμό μη-παρατηρήσιμο από το ανθρώπινο μάτι, η απόδοση ενός μοντέλου μπορεί να υποφέρει. Αυτό είναι αποτέλεσμα της αδυναμίας των μοντέλων να γενικεύουν και να μαθαίνουν αναπαραστάσεις αντικειμένων που είναι ανεξάρτητες από το πεδίο. Για την αντιμετώπιση αυτής της αδυναμίας υπάρχουν δύο κύριες προσεγγίσεις: η γενίκευση και η προσαρμογή πεδίου.

Η γενίκευση αποσκοπεί στην εκμάθηση αναπαραστάσεων που είναι ανεξάρτητες από το πεδίο, ενώ η τροποποίηση πεδίου επικεντρώνεται στην προσαρμογή ενός προεκπαιδευμένου μοντέλου σε μια κατανομή στόχο. Από τη μία, κατά την προσαρμογή πεδίου, είτε τα δεδομένα είτε οι αναπαραστάσεις τους προσαρμόζονται στις αναπαραστάσεις που έχει ήδη δει το μοντέλο. Με άλλα λόγια, είτε τα εισερχόμενα δεδομένα προσαρμόζονται στο πεδίο των δεδομένων εκπαίδευσης, είτε οι αναπαραστάσεις των αντικειμένων ταυτίζονται με αυτές του πεδίου εκπαίδευσης. Από την άλλη, αν και η γενίκευση είναι πιο δύσκολη, επιτρέπει σε ένα μοντέλο να επιτυγχάνει καλή απόδοση ακόμη και σε δεδομένα που δεν έχει ξαναδεί. Ένα μοντέλο που έχει υποστεί προσαρμογή πεδίου απαιτεί επανεκπαίδευση για κάθε νέο σύνολο δεδομένων διαφορετικού πεδίου.

Η παρούσα διατριβή διερευνά τις τεχνολογίες που χρησιμοποιούνται για σημασιολογική τμηματοποίηση και ιδίως για εφαρμογές πραγματικού χρόνου, όπως είναι η αυτόνομη οδήγηση. Με μία αρχική ιβλιογραφική ανασκόπηση, αναπτύσσονται κάποιες από τις λύσεις που προσφέρονται για την αντιμετώπιση των προκλήσεων που αναφέρονται παραπάνω. Στη συνέχεια, μελετώνται συνοπτικά κάποιες διαδεδομένες αρχιτεκτονικές που χρησιμοποιούνται για αυτόν τον σκοπό, καθώς και κάποια συνήθη σύνολα δεδομένων. Προχωρώντας, εξετάζεται μία από τις πιο σύγχρονες μεθόδους για γενίκευση πεδίου, με την χρήση της οποίας εκπαιδεύονται και αξιολογούνται κάποια από τα μοντέλα, που έχουν παρουσιατεί. Τέλος, τα μοντέλα αυτά δοκιμάζονται σε νέα πεδία γνώσης και συγκεκριμένα σε δεδομένα που προέρχονται από μη επανδρωμένα σκάφη και σε ιατρικά δεδομένα.

1.1 Σημασιολογική τμηματοποίηση με ασθενή επίβλεψη

Η δημιουργία μοντέλων τμηματοποίησης με βάση το ΝΝ αντιμετωπίζει μια σημαντική πρόκληση, καθώς η εκπαίδευση απαιτεί συνήθως σχολιασμένες εικόνες σε επίπεδο εικονοστοιχείου, μια διαδικασία έντασης πόρων. Η απόκτηση πλήρως επιβλεπόμενων δεδομένων είναι δαπανηρή και χρονοβόρα. Κατά συνέπεια, οι ερευνητές καταφεύγουν συχνά σε αδύναμους σχολιασμούς και προτείνουν μεθόδους για σημασιολογική τμηματοποίηση με ασθενή επίβλεψη, μετριάζοντας την εξάρτηση από πλήρως σχολιασμένα δεδομένα. Οι αδύναμοι σχολιασμοί, όπως τα σχολιασμένα πλαίσια οριοθέτησης, οι ετικέτες σε επίπεδο εικόνας, οι σχολιασμοί με μουτζούρες και οι σημειακοί σχολιασμοί, αποδεικνύονται πιο εφικτό να συγκεντρωθούν σε σύγκριση με τους λεπτομερείς σχολιασμούς σε επίπεδο εικονοστοιχείου. Οι ακόλουθες εργασίες κατηγοριοποιούνται με βάση τους πρωταρχικούς τύπους ετικετών με ασθενή εποπτεία.

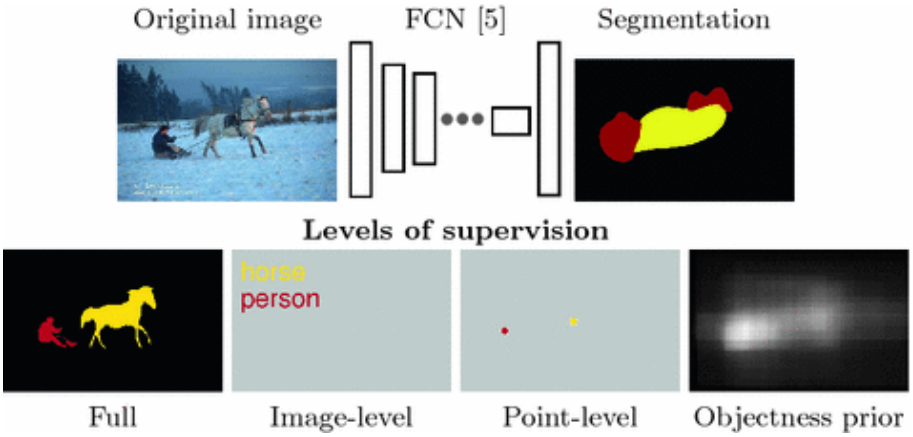

Image 1.1: *Types of weakly supervised training [1]*

1.2 Προσαρμογή τομέα στη σημασιολογική τμηματοποίηση

Τα πλήρως συνελικτικά μοντέλα έχουν επιτύχει σε εργασίες σημασιολογικής κατάτμησης. Αυτά τα μοντέλα αποδίδουν καλά σε ένα περιβάλλον με επίβλεψη, αλλά η απόδοσή τους μπορεί να μειωθεί δραστικά σε μετατοπίσεις τομέων που μπορεί να φαίνονται ήπιες σε έναν ανθρώπινο παρατηρητή. Για παράδειγμα, εάν ένα μοντέλο εκπαιδευτεί σε μια πόλη και δοκιμαστεί σε μια άλλη πόλη σε διαφορετική γεωγραφική περιοχή ή/και καιρικές συνθήκες, η απόδοση του μοντέλου μπορεί να υποβαθμιστεί σημαντικά λόγω της μετατόπισης της κατανομής σε επίπεδο εικονοστοιχείου. Η προσαρμογή τομέα είναι μια ειδική περίπτωση μάθησης μεταφοράς, η οποία χρησιμοποιεί επισημασμένα δεδομένα σε έναν ή περισσότερους συναφείς τομείς προέλευσης για την εκτέλεση νέων εργασιών στον τομέα-στόχο [28].

1.3 Αγωγός γενίκευσης τομέα

Σε αυτή την ενότητα, διερευνούμε τον αγωγό γενίκευσης τομέα (DG) που εισήγαγαν οι Hoyer et al. στην εργασία HRDA [29]. Αυτό το πρωτοποριακό πλαίσιο διαδραματίζει καθοριστικό ρόλο στην ανάπτυξη ισχυρών μοντέλων ικανών να χειρίζονται αλλαγές πεδίου, μια κοινή πρόκληση σε σενάρια αυτόνομης οδήγησης. Ο αγωγός αποτελείται από τρία βασικά στοιχεία:

- **Προ-Εκπαιδευμένοι Κωδικοποιητές**: Τα μοντέλα αξιοποιούν προ-εκπαιδευμένες ραχοκοκαλιές από το σύνολο δεδομένων ImageNet-1K [30]. Η γνώση που αποκτήθηκε κατά την προ-εκπαίδευση βοηθά στην εκμάθηση χαρακτηριστικών αμετάβλητων ως προς τον τομέα, ευθυγραμμίζοντας τις λανθάνουσες αναπαραστάσεις από το συνθετικό σύνολο δεδομένων GTA5 με εκείνες που μαθαίνονται από το ImageNet-1K, διευκολύνοντας την καλύτερη γενίκευση σε διάφορους τομείς.

- **Δειγματοληψία Σπάνιων Κλάσεων**:Για τον μετριασμό της ανισορροπίας των κλάσεων χρησιμοποιείται η δειγματοληψία σπάνιων κλάσεων. Αυτή η προσέγγιση προσαρμόζει τη στρατηγική δειγματοληψίας ώστε να δίνεται μεγαλύτερη έμφαση στις υποεκπροσωπούμενες κλάσεις (π.χ. πεζοί, πινακίδες κυκλοφορίας) που είναι λιγότερο συχνές σε

σύγκριση με τις πιο κυρίαρχες κλάσεις (π.χ. δρόμοι, κτίρια). Εστιάζοντας σε αυτές τις σπάνιες κλάσεις κατά την εκπαίδευση, το μοντέλο βελτιώνει την απόδοση σε σενάρια όπου εμφανίζονται αυτές οι κλάσεις.

- **Style-HAllucinated Dual consistEncy learning (SHADE) [31]**: Η μέθοδος SHADE είναι ένα πλαίσιο που έχει σχεδιαστεί για την ενίσχυση της γενίκευσης τομέων με την αντιμετώπιση της πρόκλησης των μετατοπίσεων τομέων, οι οποίες προκύπτουν όταν τα μοντέλα που εκπαιδεύονται σε ένα περιβάλλον δυσκολεύονται σε ένα άλλο λόγω των διαφοροποιήσεων του οπτικού στυλ. Το SHADE εισάγει συνθετικά στυλ τομέων με διαταραχές στους χάρτες χαρακτηριστικών κατά τη διάρκεια της εκπαίδευσης, επιτρέποντας στο μοντέλο να προσαρμόζεται σε ένα ευρύ φάσμα στυλ. Το πλαίσιο διασφαλίζει τη συνοχή μεταξύ της αρχικής εισόδου και της αντίστοιχης που έχει τροποποιηθεί με το στυλ, προωθώντας σταθερές προβλέψεις και βελτιώνοντας τη γενίκευση.

1.4 FarSeeFormer

Για την παρούσα διατριβή, επιλέξαμε να χρησιμοποιήσουμε μια αρχιτεκτονική που ανταποκρίνεται στις απαιτήσεις για υψηλή ακρίβεια, ανθετικικότητα και απόδοση, χρησιμοποιώντας ένα κωδικοποιητή transformer μαζί με ένα εξαιρετικά αποδοτικό συνελικτικό αποκωδικοποιητή. Ο κωδικοποιητής αποτελείται από έναν ιεραρχικό transformer, ο οποίος αυξάνει την ανθεκτικότητα του μοντέλου σε σχέση με συνελικτικές αρχιτετκονικές. Ο αποκωδικοποιητής αποτελείται από το FarSee-Net, μια συνελικτική αρχιτεκτονική πραγματικού χρόνου. Χρησιμοποιεί διαχωρίσιμες κατά βάθος συνελίξεις για μείωση του χρόνος εξαγωγής και συνελίξεις υπο-pixel για αύξηση της ανάλυσης των αποτελεσμάτων.

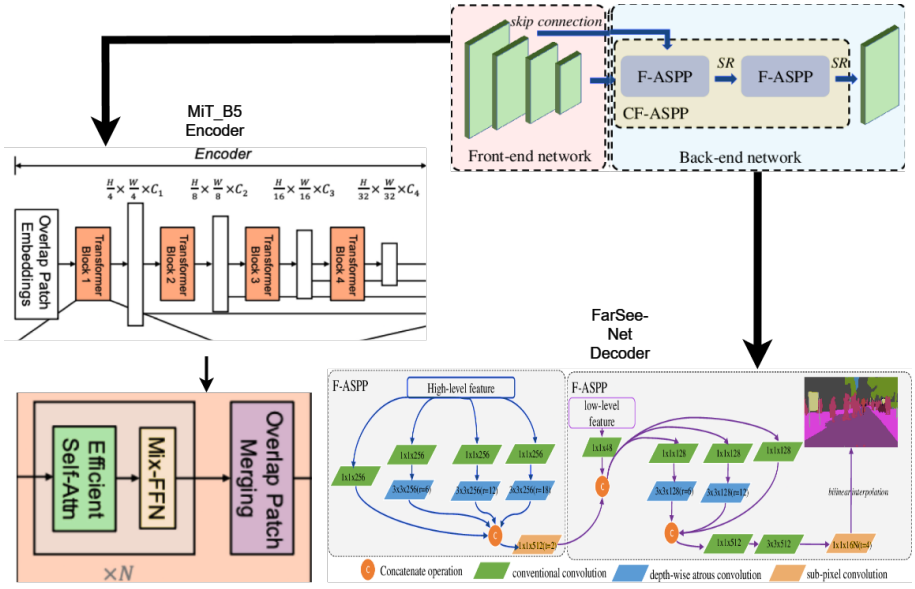

Image 1.2: *FarseeFormer*

1.5 Παρουσίαση και Ανάλυση των Αποτελεσμάτων

1.5.1 Αποτελέσματα

Μοντέλα	Δρόμος	Πεζόδρομος	Κτήριο	Τοίχος	Φράχτης	Πόλος	Φωτεινός σηματοδότης	Σημάδι	Βλάστηση	Έδαφος	Ουρανός	Άτομο	Αναβάτης	Αυτοκίνητο	Φορτηγό	Λεωφορείο	Τρένο	Μοτοσικλέτα	Ποδήλατο	**mIoU**
GTA5 → Ιτψσαπες																				
FarSee-Net	73.3	26.5	75.8	20.1	2.38	22.3	12.7	3.28	79.0	30.1	81.4	31.9	10.3	60.9	11.5	13.6	0.47	8.91	5.78	30.0
Segformer	87.7	33.0	84.8	34.1	**27.4**	35.2	**47.4**	20.5	**87.8**	42.2	86.9	65.2	35.0	88.7	45.4	46.0	21.8	**29.6**	**30.2**	49.9
DAFormer	**90.0**	**45.0**	85.4	**36.4**	26.4	**37.7**	44.7	**23.0**	87.5	**42.7**	88.0	**68.5**	39.0	**89.0**	45.1	42.5	29.5	27.7	28.3	**51.4**
FarSee-Net2	88.7	34.9	**85.6**	36.1	26.5	32.4	43.2	20.9	87.1	39.0	**88.5**	65.8	**39.6**	87.3	**46.4**	**49.7**	**36.7**	26.7	27.8	50.7

Table 1.1. *Σύγκριση μοντέλων στην κατεύθυνση της γενίκευσης τομέα. Τα μοντέλα εκπαιδεύτηκαν για 416.100 επαναλήψεις, χρησιμοποιώντας συνθετικά δεδομένα από το σύνολο δεδομένων GTA5 και αξιολογήθηκαν στο σύνολο δεδομένων ΐτψσαπες. Ο πίνακας περιέχει το IoY που επιτεύχθηκε από κάθε μοντέλο για κάθε κατηγορία.*

Μοντέλο	Μνήμη (GB)
FarSee-Net	**3.8**
Segformer	19.22
DAFormer	20.0
FarSee-Net2	17.4

Table 1.2. *Απαιτήσεις μνήμης από κάθε μοντέλο κατά την εκπαίδευση.*

1.5.2 Ανάλυση

Η εκπαίδευση και αξιολόγηση των μοντέλων διήρκεσε περίπου πέντε ημέρες για τα μοντέλα μετασχηματιστών, ενώ το συνελικτικό μοντέλο απαιτούσε περίπου δύο ημέρες. Σημαντικά, το συνελικτικό μοντέλο έδειξε σημαντικά χαμηλότερη χρήση μνήμης κατά τη διάρκεια της εκπαίδευσης, κάνοντάς το λιγότερο απαιτητικό σε πόρους. Ωστόσο, αυτή η αποδοτικότητα είχε ένα κόστος: η απόδοσή του υστερούσε σε σχέση με τα μοντέλα μετασχηματιστών, τα οποία παρουσίασαν συγκρίσιμες τιμές μέσου Διατομής (μIoY). Όλες οι κατηγορίες παρουσίασαν μια αισθητή πτώση στην ακρίβεια, ιδιαίτερα στις πιο σπάνιες κατηγορίες όπως το τρένο, το ποδήλατο και ο αναβάτης.

1.6 Συμπεράσματα και Μελλοντική Εργασία

1.6.1 Συμπεράσματα

Στην παρούσα διατριβή, πραγματοποιήθηκε ανάλυση της πρόκλησης της σημασιολογικής τμηματοποίησης για την αυτόνομη οδήγηση, καλύπτοντας διάφορες πτυχές. Εξετάστηκαν οι υπάρχουσες λύσεις στη βιβλιογραφία και παρουσιάστηκαν σημαντικά μοντέλα και σύνολα δεδομένων. Η προσέγγιση σε αυτή την εργασία πλαισιώθηκε μέσα από την οπτική της

Γενίκευσης Τομέα, με την τελική υιοθέτηση της πρωτοποριακής ροής εργασίας Γενίκευσης Τομέα που αναπτύχθηκε στα έργα των [29], [32].

Αποφασίσαμε να συγκρίνουμε τέσσερα μοντέλα: ΦαρΣεε-Νετ, Σεγφορμερ, ΔΑΦορμερ και ΦαρΣεε-Νετ2. Το ΦαρΣεε-Νετ είναι ένα από τα καλύτερα συνελικτικά μοντέλα που χρησιμοποιούνται για σημασιολογική τμηματοποίηση σε πραγματικό χρόνο. Το ΔΑΦορμερ ήταν το μοντέλο που παρουσιάστηκε στα έργα των [29], [32], ενώ το Σεγφορμερ ήταν ο πρόγονος του, που εισήγαγε τον κωδικοποιητή MiT-B5 [20]. Το ΦαρΣεε-Νετ2 είναι μια νέα αρχιτεκτονική που χρησιμοποιεί την υποδομή MiT-B5 μαζί με τον αποδοτικό και υπολογιστικά ελαφρύ αποκωδικοποιητή ΦαρΣεε-Νετ. Ανάπτυξε για αυτή τη διατριβή προκειμένου να επιτευχθούν ταχύτεροι χρόνοι παραγωγής και εκπαίδευσης.

Τα 4 μοντέλα εκπαιδεύτηκαν και δοκιμάστηκαν στα ίδια σύνολα δεδομένων (ΓΤΑ5 → ΐτψσαπες), καθώς και στα ίδια σύνολα δεδομένων που αφορούν νέους τομείς γνώσης (ΥΑ″ και ιατρική απεικόνιση). Τα αποτελέσματα έδειξαν ότι τα μοντέλα μετασχηματιστών αποδίδουν πολύ καλύτερα σε εφαρμογές του πραγματικού κόσμου, χάρη στη ροβυστισιτψ και την προσαρμοστικότητά τους. Η εξαίρεση ήταν τα ιατρικά δεδομένα, όπου το συνελικτικό μοντέλο παρέμεινε ανταγωνιστικό, υποθέτοντας λόγω της μετατόπισης τομέα μεταξύ των δεδομένων εκπαίδευσης και αξιολόγησης να μην είναι τόσο έντονη. Το ΦαρΣεε-Νετ2 υπερέβη τα άλλα μοντέλα στο σύνολο δεδομένων ΥΑ″ και τα πήγε συγκρίσιμα με τα καλύτερα μοντέλα σε καθένα από τα άλλα 2 πειράματα. Το πλεονέκτημα του ΦαρΣεε-Νετ2 βρίσκεται στη μικρότερη υπολογιστική απαίτηση κατά την ανάπτυξη για δοκιμές και εκπαίδευση.

1.6.2 Μελλοντική Εργασία

Ενώ οι μετασχηματιστές είναι ροβυστ και ανθεκτικοί σε μετατοπίσεις τομέα, απέχουν πολύ από το να είναι βιώσιμη λύση σε καθήκοντα πραγματικού χρόνου. Στην περίπτωση μας, χρησιμοποιώντας έναν αποδοτικό αποκωδικοποιητή, καταφέραμε να επιταχύνουμε την ταχύτητα παραγωγής χωρίς να θυσιάσουμε την ακρίβεια. Ωστόσο, αυτή η ελαφρά μείωση στον χρόνο παραγωγής δεν είναι αρκετή για να επιτευχθεί τμηματοποίηση πραγματικού χρόνου ή ταχύτητες συγκρίσιμες με εκείνες των συνελικτικών μοντέλων. Προτείνουμε να καταβληθεί προσπάθεια για να μειωθεί το υπολογιστικό βάρος που επιβάλλουν τα μοντέλα μετασχηματιστών, καθώς φαίνεται να είναι η κύρια πηγή της αύξησης των χρόνων παραγωγής σε σύγκριση με τις συνελικτικές αρχιτεκτονικές.

Chapter 2

Introduction

In recent years, artificial intelligence (AI) has experienced remarkable growth, transforming a wide range of scientific disciplines. This rapid progress is largely driven by advances in computational power and the availability of vast amounts of data for training AI models. Among the key outcomes of these advancements is the rise of deep learning, a fundamental technology in modern AI applications.

One of the fields most profoundly impacted by deep learning is Computer Vision. With AI's evolving capabilities, deep learning models have become highly effective at analyzing complex data with remarkable precision. These models are now integral to a variety of real-time applications, including one of the most critical in the domain of autonomous driving: semantic segmentation. Semantic segmentation enables autonomous vehicles to gain a detailed, pixel-level understanding of their environment, allowing them to make informed decisions based on real-time data from high-resolution cameras.

For such applications, ensuring that AI models operate in real time with high accuracy is essential. Significant strides have been made in developing efficient architectures capable of processing multiple high-resolution frames while maintaining robust performance. However, one of the major challenges in this field is ensuring that these models are resilient to varying environmental conditions, which requires access to diverse and abundant data. Unfortunately, obtaining such data is not always feasible, and model performance can degrade when faced with conditions that differ from the training data.

Today, numerous datasets for autonomous driving exist, many of which offer diverse data—including adverse scenarios—that can support the development of more resilient models. These datasets can be either real-world or virtual, and selecting the right dataset is crucial for training deep learning models that can generalize effectively to real-world applications. However, if the data distribution in the training set does not align with the real-world conditions in which the model is deployed, the model may struggle to generalize, resulting in poor performance.

This challenge has given rise to research in domain adaptation and domain generalization. In domain adaptation, a pre-trained model is fine-tuned to better align with the target domain, either by adjusting the data distribution or by refining object representations across domains. In contrast, domain generalization aims to develop models that perform well in a target domain without requiring further training on that domain's data. These approaches are particularly relevant in the context of semantic segmentation for

autonomous driving, where models must adapt to a wide variety of environments.

In conclusion, while significant progress has been made in addressing the challenges of semantic segmentation for autonomous vehicles, there is still much to explore. Understanding and improving the generalization capabilities of these models remains a key area of ongoing research, offering exciting potential for future advancements in autonomous driving technology.

2.1 Objective of the Thesis

The primary objective of this thesis is to explore the datasets and techniques employed in addressing the task of semantic segmentation within the context of autonomous driving. Additionally, it will review several state-of-the-art models currently used in semantic segmentation. A domain generalization approach will be applied to test and compare the performance of these models. Finally, the models will be extended beyond the autonomous driving domain, with experiments conducted on datasets from other fields of knowledge.

2.2 Thesis Organization

This thesis is organized into seven chapters. Chapter 1 provides an introduction to the thesis, outlining its scope and objectives. Chapter 2 presents the theoretical background, starting with an overview of the fundamental architectures used in image processing, and concluding with a discussion of the challenges of domain generalization. Chapter 3 reviews weakly supervised training approaches and relevant work on domain adaptation. In Chapter 4, methods for accelerating semantic segmentation are discussed, along with a review of some state-of-the-art architectures. Chapter 5 presents some of the most common datasets used in semantic segmentation for autonomous driving. Chapter 6 details the datasets and training pipeline employed in the experiments. Chapter 7 presents the experimental results and analysis. Finally, Chapter 8 summarizes the contributions of this thesis and suggests potential directions for future research.

Part I

Theory

Chapter 3

Theoretical Background

In this chapter, the theoretical background necessary for understanding this work will be presented. Section 2.1 explores the history of deep neural networks and the evolution of architectures used for image segmentation over the years. Section 2.2 provides an overview of the different types of image segmentation, while Section 2.3 discusses the domain generalization challenges encountered in semantic segmentation for autonomous driving.

3.1 Deep Neural Networks

In order to better understand the concepts and work presented, it is important to have a grasp of Neural Networks and Deep Learning Models. Therefore, it is appropriate to provide an overview of the main theory behind Neural Networks, as well as some information on the primary architectures used for image processing.

3.1.1 Neural Networks

Neural networks are computational models inspired by the structure and function of the human brain. They are made up of interconnected nodes called "neurons," that process input data and learn patterns through training. Each neuron receives inputs, applies weights, sums them, and passes the result through an activation function to produce an output. These networks are mainly used for tasks such as classification, regression, and pattern recognition [33].

Neural networks have been around since the 1940s, with the Perceptron being one of the first models. However, they only became popular in the 1980s with the development of backpropagation. In the 2010s, advancements in hardware and the availability of large datasets led to breakthroughs in deep learning, sparking a renewed interest in neural networks.

3.1.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) were created to handle the growing complexity of computer vision tasks. They are the foundational architecture for computer vision models and are specialized deep neural networks designed primarily for processing structured

grid data, such as images. Unlike networks that relied on densely connected layers, which became unwieldy due to their excessive connectivity, CNNs share parameters across the spatial dimensions of the input, reducing the number of parameters compared to fully connected networks. Additionally, CNNs automatically learn hierarchical features, starting from low-level features (like edges) in early layers to high-level features (like shapes and objects) in deeper layers. Along with their local receptive fields, they recognize patterns regardless of their position in the input image, providing robustness against spatial translations.

The main components of CNNs are:

- Convolutional Layers
- Pooling Layers
- Classification Fully Connected Layers
- The Output Layer

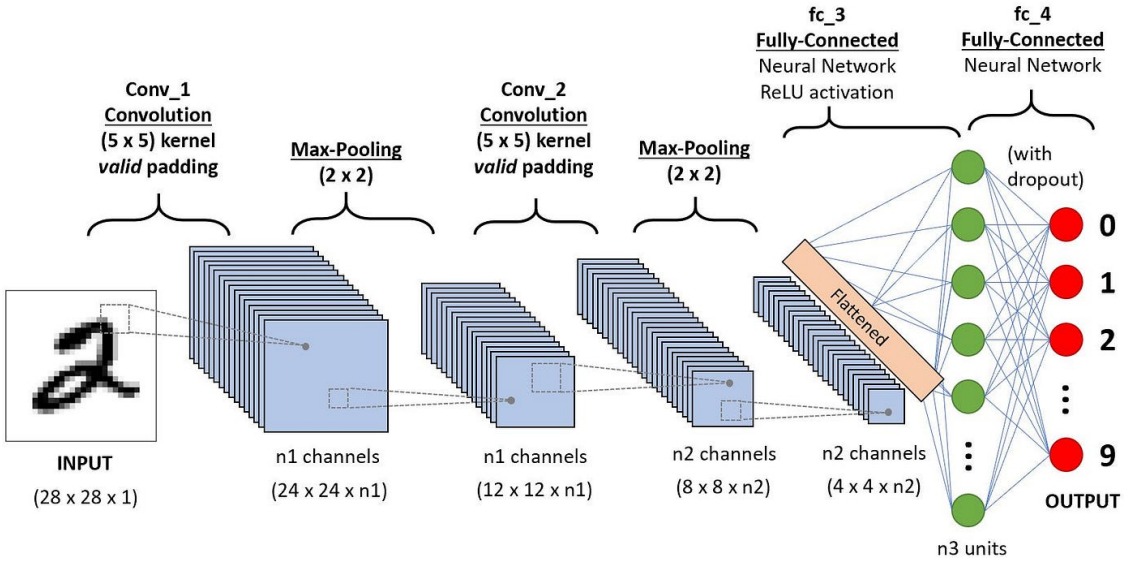

Image 3.1: *An example of CNN architecture [2]*

The most important and unique parts of a CNN are the Convolution Layers and the Pooling Layers. The functionality and usage of those types of layers are analyzed below.

Convolution Layer (or Kernel)

The Convolutional Layers contain convolutional filters that execute the following algorithm. Each kernel slides over the input image in a process called convolution, performing an element-wise multiplication between the kernel and a subset of the input (often referred to as the receptive field). The results of these multiplications are then summed to produce a single value. This operation is repeated as the kernel moves across the image, generating a feature map or activation map that represents the presence of specific

patterns in different regions of the input. Some spatial information might be lost in the process, but a lot of new information is derived.

The convolution operation for an input image I and a kernel K is defined as:

$$(I * K)(x, y) = \sum_m \sum_n I(x + m, y + n) \cdot K(m, n)$$

where $I(x, y)$ represents the pixel value of the input image at position (x, y), and $K(m, n)$ is the value of the kernel at position (m, n). The summation is carried out over the dimensions of the kernel.

Two important hyperparameters that control the behavior of the convolution operation are stride and padding. The stride defines the step size of the kernel as it moves across the input image. A stride of 1 means the kernel shifts by one pixel at a time, while a larger stride results in downsampling, reducing the spatial resolution of the output feature map. Padding refers to adding extra pixels (usually zeros) around the border of the input image. It allows the kernel to process edge pixels more effectively and helps control the size of the output feature maps. Finally, in some cases, convolutional layers have more than one filter in parallel. The number of filters in parallel should match the number of channels of the input. Image 3.2 presents an example of the aforementioned convolution.

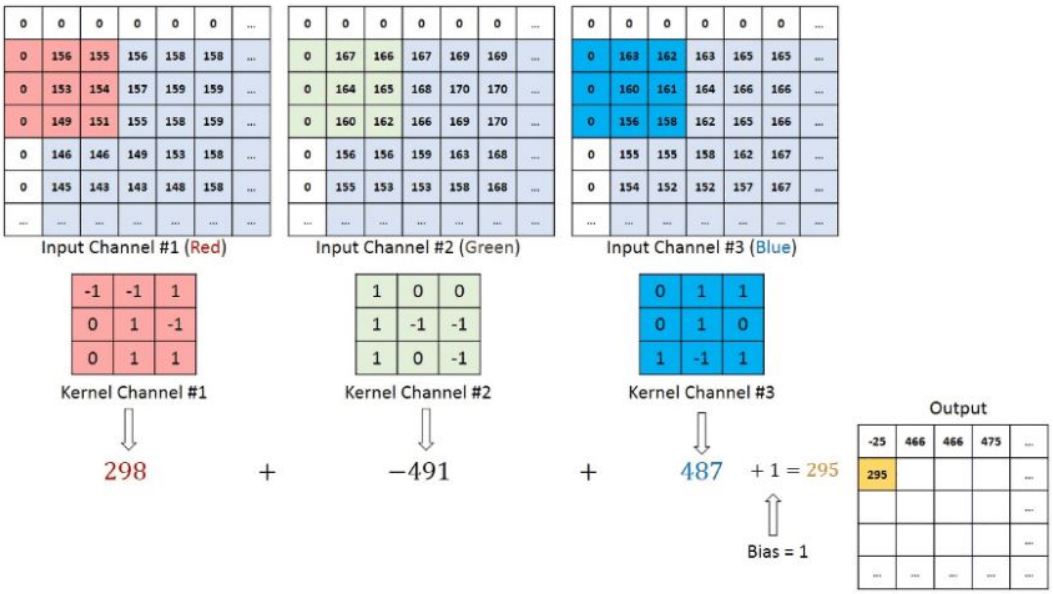

Image 3.2: *An example of a convolution operation [3]*

Pooling Layer

Pooling layers work in a way similar to convolution layers. A filter is sliding across the grid of the input, but instead of applying a convolution between the input data and the kernel, the values of the window are replaced with a number. Specifically, there are two types of pooling-layers:

- **Max Pooling** returns the maximum value of the elements in each sub-matrix.

- **Average Pooling** returns the average value of the elements in each sub-matrix.

A pooling layer serves the function of progressively reducing the spatial dimensions of the feature maps produced by convolutional layers. This downsampling process preserves important features while reducing computational complexity, memory usage, and the likelihood of overfitting. Pooling layers introduce an element of spatial invariance, allowing the network to maintain robustness to small translations, rotations, or distortions in the input data.

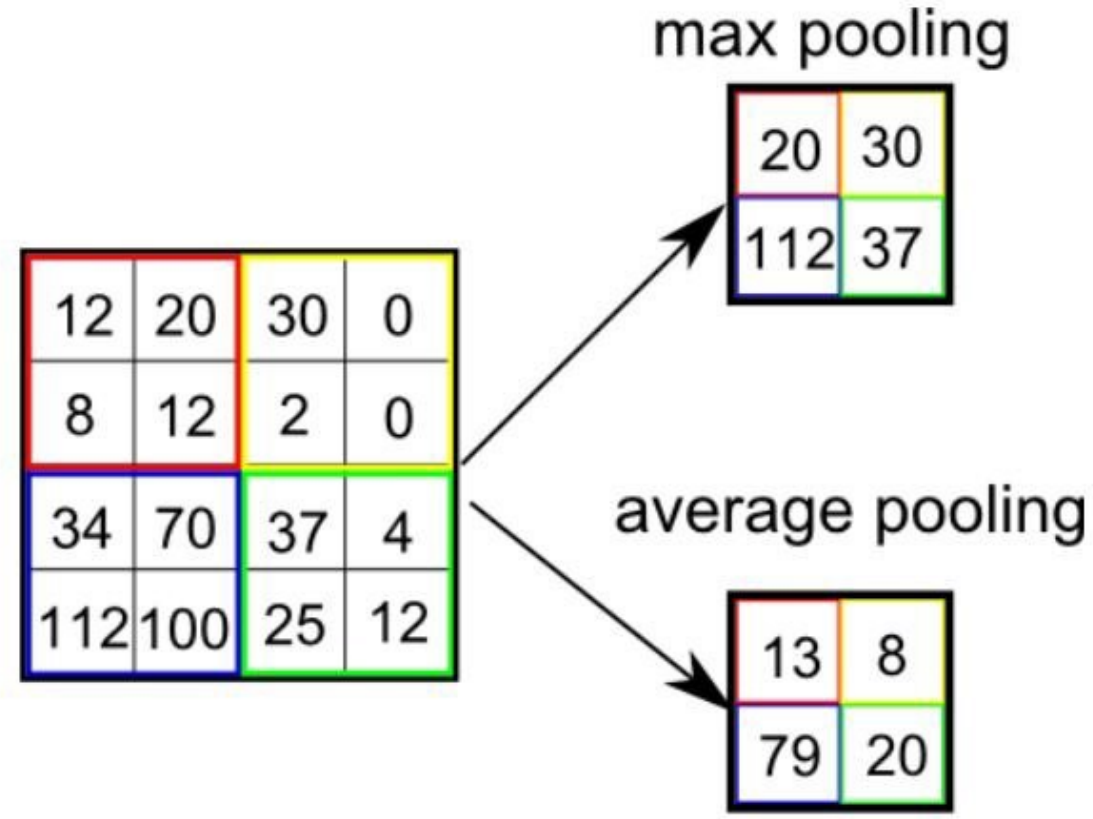

Image 3.3: *An example of a Pooling Layer operation [3]*

Fully Convolutional Neural Networks

Fully Convolutional Neural Networks (FCNNs) are a type of neural network designed for tasks that involve pixel-wise predictions, such as semantic segmentation. In semantic segmentation, each pixel of an input image is assigned a class label. Unlike traditional Convolutional Neural Networks (CNNs) that use fully connected layers for classification tasks, FCNNs consist entirely of convolutional layers and do not have any fully connected layers. This structure allows FCNNs to take input images of any size and produce spatially consistent predictions, making them well-suited for dense prediction tasks. Some key characteristics of FCNNs are the absence of fully connected layers and their input flexibility.

In traditional CNNs, fully connected layers are typically used after convolutional layers to perform high-level reasoning and final classification. In FCNNs, these fully connected layers are replaced by convolutional layers, enabling the network to produce spatial maps

instead of single-class predictions. This feature allows FCNNs to retain spatial information throughout the network, which is crucial for tasks like image segmentation.

FCNNs can handle inputs of varying sizes because their convolutional layers do not require a fixed input size. In contrast, fully connected layers in traditional CNNs demand a fixed input size. The absence of these layers in FCNNs allows for flexibility in handling inputs of different dimensions, making the architecture highly adaptable for real-world applications.

FCNNs are widely used and effective, but they do have some limitations. One key issue is that they cannot process images in real time, which can be a problem for tasks requiring immediate results. Additionally, FCNs struggle to capture all the necessary information from an image, especially the broader context that aids in accurate segmentation. Moreover, they are challenging to implement on three-dimensional images, which limits their usefulness in certain situations.

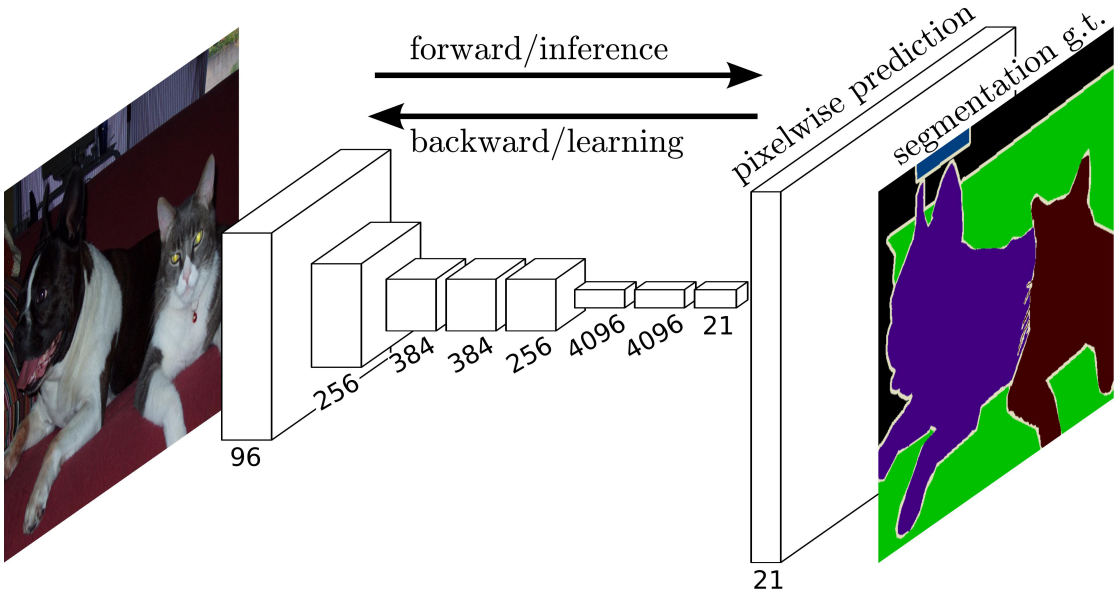

Image 3.4: *An example of a Fully Convolutional Neural Network [3]*

3.1.3 Encode-Decoder Architectures

Encode-decoder models are models designed to handle tasks that require transforming one type of data into another, often used in areas like image segmentation, machine translation, and speech recognition. The architecture consists of two primary components: an encoder that compresses the input data into a compact representation, and a decoder that reconstructs the output from this representation.

The encoder processes input data and extracts important features. It usually includes a series of layers, such as convolutional layers for computer vision or recurrent layers for sequence-based tasks. These layers progressively downsample and compress the input into a lower-dimensional representation, often referred to as the latent space or bottleneck. The output of the encoder is a concise feature map or vector containing the essential information of the input data in a more abstract form.

The decoder takes the encoded representation and tries to reconstruct the output in a specific format, such as a segmented image in computer vision or a translated sentence in natural language processing. The decoder typically follows the structure of the encoder but in reverse, using techniques like upsampling or transposed convolutions to restore the original resolution or structure of the data.

Encoder-Decoder architectures are a powerful framework for solving tasks that involve transforming input data into structured outputs. These models work by compressing input data into a latent representation and then reconstructing it in the desired format. They have proven to be highly effective in fields like image segmentation, machine translation, and speech recognition. Their ability to handle variable input and output lengths, as well as maintain spatial details through skip connections, makes them well-suited for complex tasks requiring high-level abstraction and detailed reconstruction.

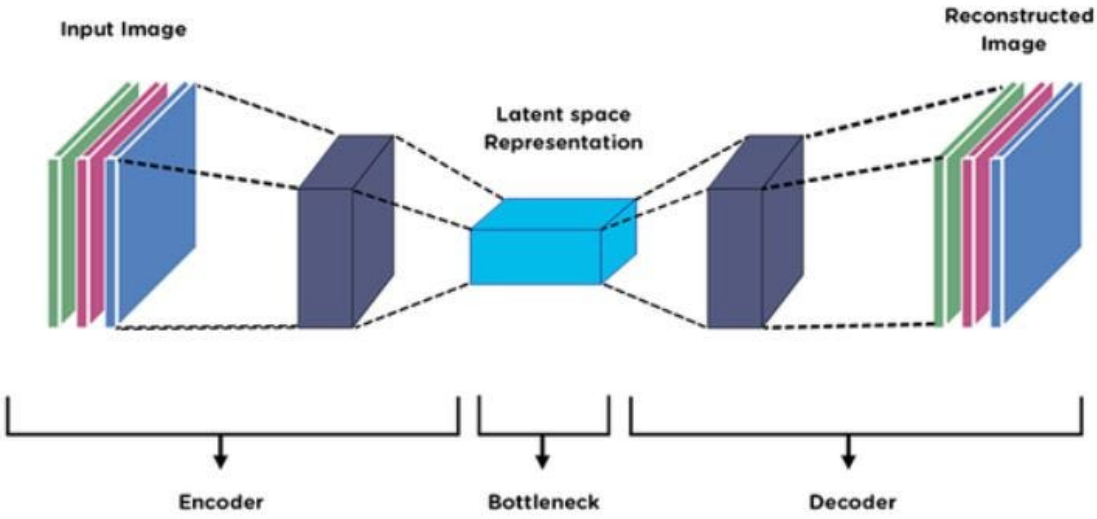

Image 3.5: *An example of the basic architecture of an Encoder Decoder Network [4]*

3.1.4 Vision Transformers

Convolutional networks, such as CNNS or FCNS, are commonly used for semantic segmentation, but they have limitations. For instance, the final output segmentation image of the feature map has low resolution and they are not effective at capturing long-range dependencies of the feature maps. The emergence of Vision Transformer (ViT), inspired by transformer-based architectures in NLP, has shown promising results in addressing these limitations.

A Transformer in machine learning is a deep learning model that utilizes self-attention mechanisms. Self-attention allows the model to weigh the importance of different parts of the input during processing, enabling it to capture dependencies between parts more flexibly than Convolutional Neural Networks. However, applying self-attention in images is computationally expensive due to the quadratic cost resulting from each pixel attending to every other pixel.

Dosovitskiy et al. [5] proposed a novel approach for handling images by dividing them into patches and treating them as tokens, similar to NLP. Instead of pixel-wise attention, they implemented patch-wise attention, reducing computational complexity compared to applying self-attention to convolutional architecture. To account for transformers lacking the inherent inductive bias of CNNs for capturing spatial relationships, they added position embeddings to the patch embeddings, maintaining information about the relative positions of patches in the image.

The core of the Vision Transformer [5] is a stack of transformer encoder blocks, each consisting of multi-head self-attention and feed-forward layers. The self-attention mechanism allows the model to learn relationships between different patches, capturing both local and global dependencies across the entire image. A special token called the classification token (CLS) is added to the input sequence of patch embeddings. After passing through the transformer layers, this token accumulates information from the entire image, and its final representation is used for classification.

The final layers of Vision Transformers differ based on the model's task. For classification models, the final layer can be a Multi-Layer Perceptron Head. This layer takes the final vector representation of the input image and outputs the probability of each class. For segmentation-oriented models, the final part of the model is a Decoder, similar to Encoder-Decoder architectures. This decoder outputs a matrix with the probabilities of each class for each pixel of the original input image.

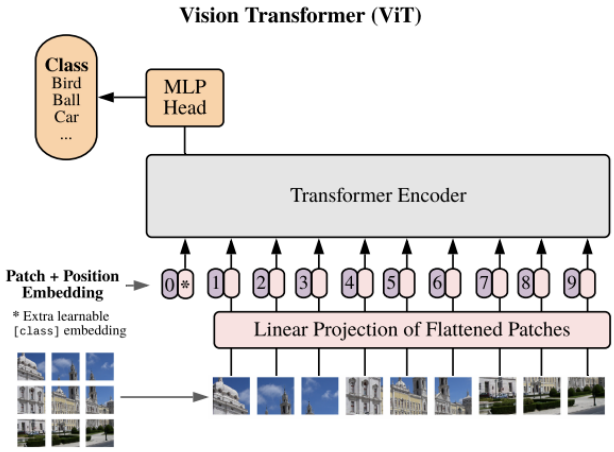

Image 3.6: *Architectur of VIT [5]*

3.2 Domain Generalization and the Out-of-Distribution Problem

Machine learning models, including those used for semantic segmentation in autonomous driving, aim to learn patterns from training data to make predictions on unseen data. However, real-world applications often expose models to environments with statistical characteristics vastly different from their training data. This mismatch, referred to

as the Out-of-Distribution (OOD) Problem, occurs when models encounter unseen data that does not align with the training distribution, severely impacting performance.

3.2.1 Domain Shift

Domain shift is a major cause of the OOD problem. It refers to changes in the distribution of data between the training (source) and testing (target) domains. Domain shift manifests in various ways in autonomous driving:

- **Environmental Changes**: Variations in weather (fog, rain, snow), lighting conditions (day vs. night), and seasons (winter vs. summer) can drastically alter the appearance of the scene.
- **Geographical Differences**: Models trained in one geographic location (e.g., urban city) may struggle to generalize to others (e.g., rural areas) due to differences in road layouts, vegetation, or infrastructure.
- **Sensor Variability**: Autonomous vehicles may use different cameras, LiDARs, or radar sensors, each with distinct resolutions and noise patterns, leading to a distribution mismatch if the model wasn't trained on data from those specific sensors.

3.2.2 Dataset Bias and Overfitting

Another contributor to the OOD problem is dataset bias—when the training data fails to capture the full diversity of real-world scenarios. If the training data is overly specific to a certain environment (e.g., clear weather, daytime), the model will likely fail when exposed to unseen conditions. Additionally, overfitting occurs when a model learns spurious correlations in the training data. These correlations, while effective during training, do not hold in different or novel contexts, exacerbating the OOD problem.

3.2.3 Domain Generalization (DG)

Domain Generalization is an approach that aims to mitigate the OOD problem by training models capable of generalizing to new, unseen domains. In DG, models learn domain-invariant features, focusing on patterns that are robust to changes in environmental factors and sensor variations. For autonomous driving, this is essential for ensuring consistent performance across various weather conditions, geographic locations, and camera configurations.

Challenges in domain generalization include:

- **Complexity of Real-World Variations**: The wide range of potential domain shifts (weather, lighting, urban vs. rural) makes it difficult for models to generalize across all possible environments.
- **Lack of Target Domain Data**: Unlike domain adaptation, which assumes access to some data from the target domain, domain generalization requires robustness without prior exposure to target domain samples.

By addressing these challenges, models for semantic segmentation in autonomous driving can improve robustness, ensuring safe and reliable operation even in diverse and unseen environments.

Chapter 4

Weakly-supervised semantic segmentation and Domain Adaptation

In this chapter we are going to present some methods used for weakly-supervised semantic segmentation and domain adaptation. In section 4.3 we will showcase the domain generalization pipeline that will be used in the experiments of chapter 7.

4.1 Weakly-supervised semantic segmentation

Building CNN-based segmentation models faces a significant challenge where training typically necessitates pixel-level annotated images, a resource-intensive process. The acquisition of fully supervised data is both costly and time-consuming. Consequently, researchers often resort to weak annotations and propose methods for weakly supervised semantic segmentation, mitigating the reliance on fully annotated data. Weak annotations, such as annotated bounding boxes, image-level labels, scribble annotations, and point annotations, prove more feasible to gather compared to detailed pixel-level annotations. The following papers are categorized based on the primary types of weakly supervised labels.

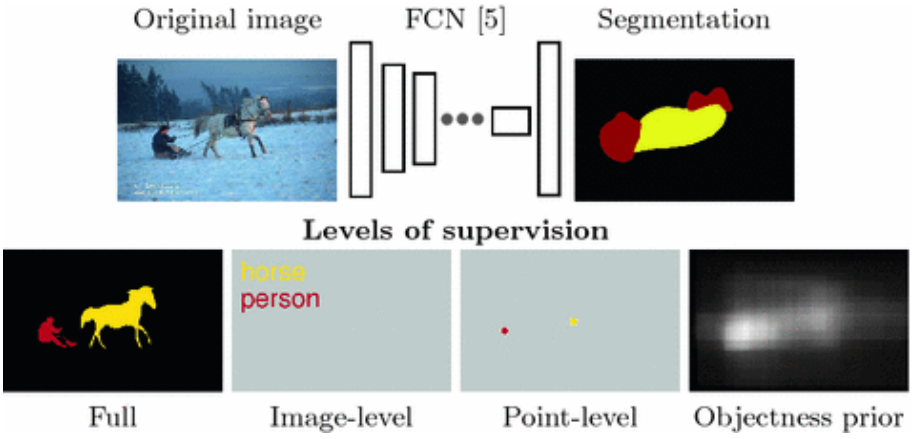

Image 4.1: *Types of weakly supervised training [1]*

4.1.1 Segmentation algorithm based on image-level labels

The main paradigm of using image-level labels to do semantic segmentation tasks is first generating the score map or heat map from the pretext task, such as the standard classification task, as the original rough mask. Then the researchers apply some clustering algorithms to refine and improve the generating mask iteratively until getting a satisfactory result. The last procedure is to feed the mask produced from the previous step as the fully annotated label to some pre-defined models that do standard supervised segmentation tasks. The annotator only needs to say whether or not a particular object class appears in an image, not how many of them there are.

Extensive research has been conducted on image-level labels. The focus is to improve the use of these labels to enhance class activation maps, which are crucial in obtaining accurate rough masks [6], [34], [35]. Some methods use the image-level label as a constraint to the network, which encourages the output to follow a latent probability distribution in the constraint manifold [36].

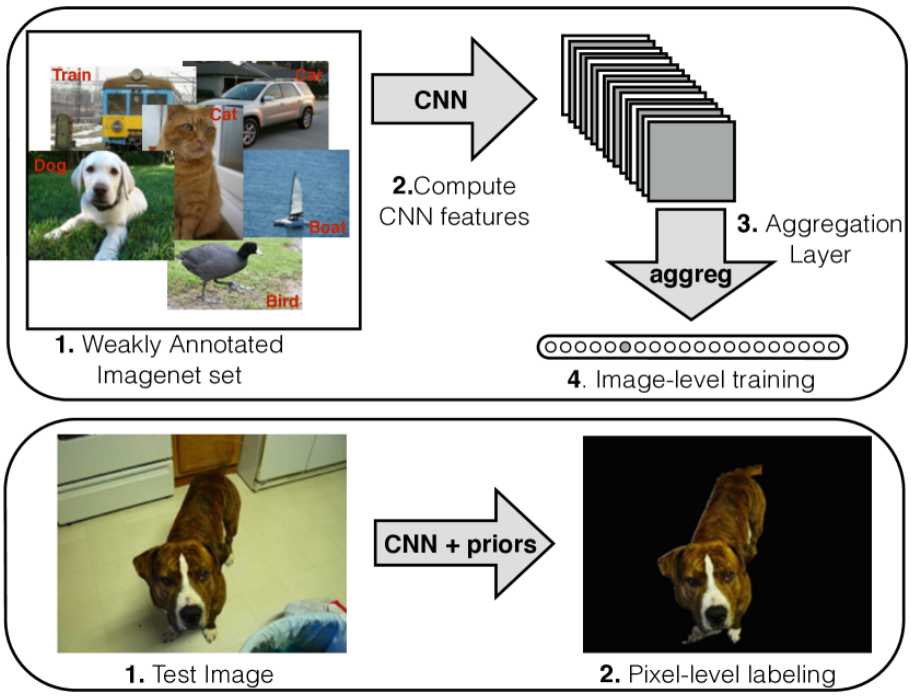

Image 4.2: *An example of utilizing image-level labels [6]*

4.1.2 Segmentation algorithm based on bounding-box

These methods generate rough segmentation maps of the target using bounding-box labels and then optimize the model iteratively. However, it relies on image annotation quality compared to the image-level label method. The research conducted using this kind of annotation, uses the bounding boxes as a basis, to try and acquire the information regarding the shape of the object placed inside the box and then try to improve the segmentation mask produced through iterative training [7], [37], [38].

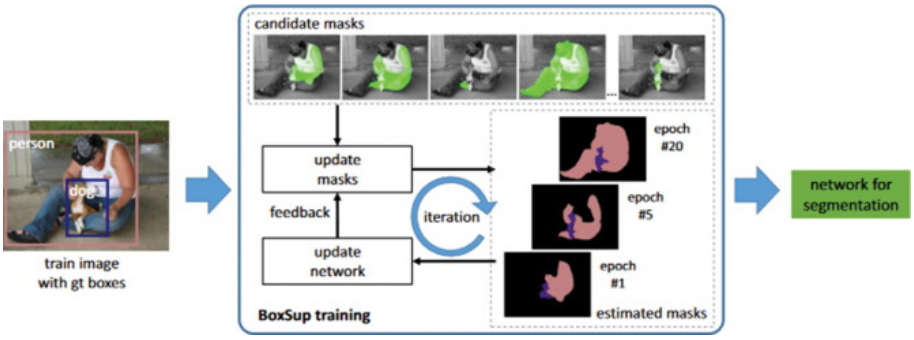

Image 4.3: *The BoxSup model. [7]*

4.1.3 Segmentation algorithm based on point

Referring to an object by pointing is the most natural way for humans, as in "That cat over there" or "What is that over there?". The technique has been proven to be useful in various fields, including robotics and human-computer interaction. However, it has not been widely used in semantic segmentation. Recently, some researchers have proposed new point-based semantic segmentation methods that incorporate point supervision into the training loss function. These methods assign only one label point to each class.

In certain research, annotated points are utilized to exploit semantic relationships. This is achieved by promoting consistency in feature representations of intra- and inter-category points. Essentially, points within the same category should have more similar feature representations than those from different categories, even across different training images [39]. Other methods use point labels and generic priors to assign probabilities of pixel classification for object separation [1].

4.2 Domain adaptation in semantic segmentation

Fully convolutional models have been successful for semantic segmentation tasks. These models perform well in a supervised setting, but their performance can drastically reduce under domain shifts that may appear mild to a human observer. For instance, if a model is trained on one city and tested on another city in a different geographic region and/or weather condition, the model's performance may degrade significantly due to the pixel-level distribution shift. Domain adaptation is a specific case of transfer learning, which utilizes labeled data in one or more related source domains to perform new tasks in the target domain [28].

4.2.1 Input-level domain adaptation

Many computer vision tasks require the translation of images while maintaining the class-related features of the original images. Style transfer methods are often used for this purpose, but the quality of the generated images can be an issue. Even minor issues at the pixel level can significantly affect the accuracy of semantic segmentation models. To overcome this, several studies have focused on ensuring semantic consistency during the image translation process. This can enhance the quality of the generated images. In this section, we will categorize these methods into two groups: GAN-based (generative adversarial network) methods and style transfer methods that use various techniques to translate images.

GAN-based methods utilize generative models to perform the style transfer of the original photos to the target domain and enable the segmentation model to perform the task. The two steps are set in a bidirectional closed-loop learning framework for domain adaptation of image semantic segmentation [8]. One such work is showcased in image 4.4.

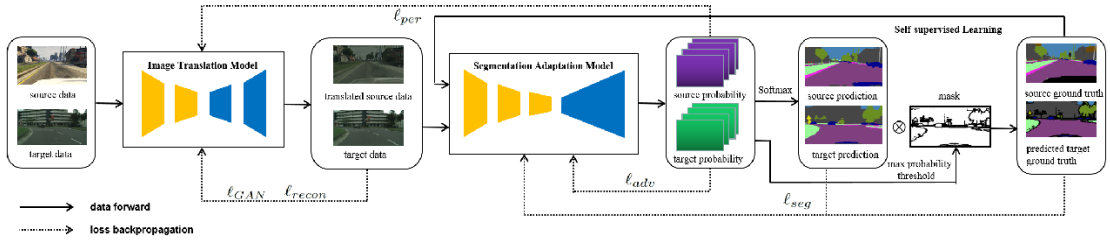

Image 4.4: *Bidirectional Learning for Domain Adaptation of Semantic Segmentation [8].*

In contrast to GAN-based methods, which are computationally intensive, style transfer methods utilize traditional neural style transfer techniques to achieve similar results. Yang et al. [40] have proposed a spectral transfer method based on the Fourier Transform that does not require any training. This method swaps the low-frequency component of the source images' spectrum with that of the target images. This way, the translated image is mapped to the target style without any change in semantic content. Wu et al. [41] employed an image generator to align the distributions of mean and variance of feature maps between the source and target domains at the pixel level. This is because these statistics are easy to optimize and provide sufficient information for achieving good stylization.

4.2.2 Feature-level domain adaptation

Possible solutions to address domain shifts involve aligning the distributions of feature latent embeddings. One way to achieve this is by modifying the feature extractor to generate domain-invariant features. By changing the distribution of latent representations between source and target domains, the network classifier can learn to segment both representations from the same latent space. This can be achieved by relying solely on the supervision from source data [42].

Some research has achieved domain invariance by using a conservative loss, enabling the network to learn discriminative features that are invariant to domain changes through

gradient ascent [43]. Other methods add extra networks and losses to regularize the features extracted by the backbone encoder network [44]. Finally, some approaches align the distributions of activations of intermediate layers, rather than only matching the output distributions of the source and target domains [45].

4.2.3 Output-level domain adaptation

Output-level domain adaptation involves modifying a model's predictions to better match the target domain. This method is particularly useful when input features are similar between the source and target domains but there are differences in the output distribution, such as labels or predictions. The goal of domain adaptation is to make the model's output consistent with the characteristics of the target domain. Some researchers use adversarial learning techniques to force the segmentation output to be in the target domain [46], [47], [48], [49]. Others use loss functions that prevent the training process from being dominated by easy-to-transfer samples in the target domain [50].

4.3 Domain Generalization Pipeline

In this section, we explore the domain generalization (DG) pipeline introduced by Hoyer et al. in the HRDA paper [29]. This cutting-edge framework plays a pivotal role in developing robust models capable of handling domain shifts, a common challenge in autonomous driving scenarios. The pipeline consists of three essential components:

- **Pre-trained backbones**: The models leverage pre-trained backbones from the ImageNet1K dataset [30]. The knowledge acquired during pretraining aids in learning domaininvariant features by aligning latent representations from the synthetic GTA5 dataset with those learned from ImageNet-1K, facilitating better generalization across domains.

- **Rare Class Sampling**: To mitigate class imbalance, rare class sampling is employed. This approach adjusts the sampling strategy to give more prominence to underrepresented classes (e.g., pedestrians, traffic signs) which are less frequent compared to more dominant classes (e.g., roads, buildings). By focusing on these rare classes during training, the model improves performance in scenarios where these classes occur.

- **Style-HAllucinated Dual consistEncy learning (SHADE) [31]**: SHADE is a framework designed to enhance domain generalization by tackling the challenge of domain shifts, which arise when models trained in one environment struggle in another due to visual style variations. SHADE introduces synthetic domain styles by perturbing feature maps during training, allowing the model to adapt to a diverse range of styles. The framework ensures consistency between the original input and its style-altered counterpart, promoting stable predictions and improving generalization.

Chapter 5

Real-time semantic segmentation

Semantic segmentation models based on deep learning have achieved impressive accuracy in recent years. However, for applications such as autonomous driving, efficiency and reduced inference time are crucial. In this section, we will present all the existing approaches that can be used in deep neural network architecture design to achieve faster response times in semantic image segmentation models.

5.1 Convolution Factorization—Depthwise Separable Convolutions

Convolutional layers are a crucial part of most deep-learning models. Therefore, making the convolutional operations performed in the network's layers more computationally efficient can significantly improve the model's speed performance.

One popular design choice for improving convolutions is the use of depthwise separable convolutions, which is a type of factorized/decomposed convolutions [51]. The depthwise separable convolution method divides the computation process into two distinct steps. First, a single convolutional filter is applied per each input channel (depthwise convolution), and then a linear combination of the output of the depthwise convolution is considered through a pointwise convolution.

The equation gives the ratio of computational complexity between depthwise separable convolutions and standard convolutions: Ratio = $1/N + 1/D^2$, where N is the number of filters of size D×D.

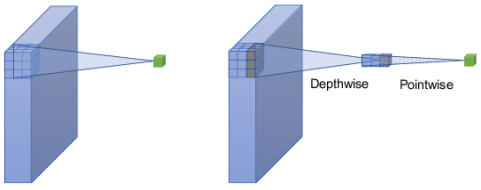

Image 5.1: *(a) Standard convolution. (b) Depthwise separable convolution [9]*

5.2 Channel Shuffling

In standard group convolution, each input channel is associated with only one output channel. However, in the case of channel shuffling, a group convolution takes data from different input groups, and every input channel will correlate with every output channel. To achieve this, we can divide the channels in each group into several subgroups and then feed each group in the next layer with different subgroups. This can be implemented efficiently by a channel shuffle operation [10].

To illustrate, imagine a convolutional layer with g groups where the output has $g \times n$ channels. We first reshape the output channel dimension into (g, n), transpose it, and then flatten it back as the input of the next layer. The channel shuffle operation is differentiable, meaning that it can be embedded into network structures for end-to-end training.

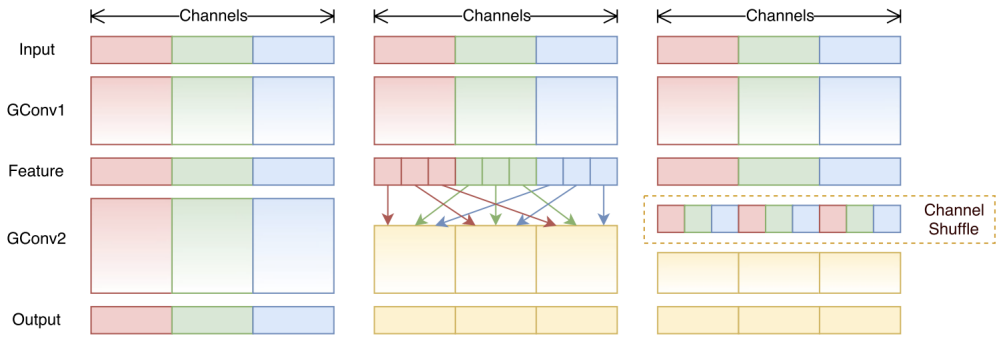

Image 5.2: *Showcase of channel shuffle (adapted from [10])*

5.3 Early Downsampling

Processing large input frames can be very expensive. One solution to this is to downsample the frames in the early stages of the network, using only a small set of feature maps. The initial network layers should focus on feature extraction and preprocessing of the input data for the following parts of the architecture, rather than contributing to the classification stage. This approach is used by the ENet model architecture [52] to prevent spatial information loss due to downsampling.

ENet's model architecture is based on the SegNet [11] approach, which saves indices of elements chosen in max-pooling layers, and uses them to produce sparse upsampled maps in the decoder. This approach reduces memory requirements while recovering spatial information. However, it is not recommended for applications where the initial image contains fine image details that have the potential to disappear after the corresponding max-pooling operation.

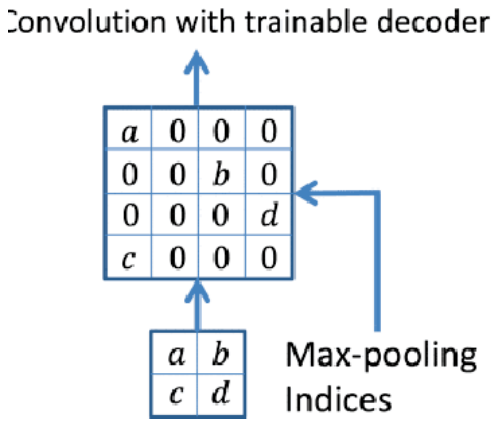

Image 5.3: *SegNet decoder uses the max-pooling indices to upsample the feature maps [11]*

5.4 The Use of Small Size Decoders

To simplify the architecture of an encoder-decoder model and reduce computational costs, one can reduce the size of the decoder [52]. This approach is based on the idea that the encoder should process input data with a smaller image resolution. Meanwhile, the role of the decoder is solely to perfect the details of the output image by upsampling the encoder's output. Therefore, reducing the size of the decoder is a cost-effective solution. This method is generally effective as the reduction in the decoder's size doesn't usually impact its effectiveness.

5.5 Efficient Reduction of the Feature Maps' Grid Size

To reduce the size of feature maps, pooling operations are commonly applied. However, these operations can create representational bottlenecks in the network filters. This can be avoided by increasing the activation dimension of the filters, which leads to increased computational costs. To address this issue, Szegedy et al. [53] suggested a pooling operation that involves performing a convolution of stride 2 in parallel, followed by concatenation of the resulting filter banks. Many such approaches exist that have been shown to reduce feature map size and improve efficiency, while achieving state-of-the-art effectiveness [54].

5.6 Increasing Network Depth While Decreasing Kernel Size

Using small (3x3) convolutional filters has been shown to improve the standard configurations of CNNs. With smaller filters, the network can be made deeper by adding more convolutional layers while reducing the number of parameters. This technique not only reduces computational cost but also increases the accuracy of the network [55], [56].

5.7 Two-Branch Networks

Two-branch networks have been developed to address the trade-off between accuracy and inference time. One branch is responsible for capturing spatial details and generating a high-resolution feature representation, while the other branch obtains high-level semantic context. These networks achieve a beneficial balance between speed and accuracy by using one pathway as a lightweight encoder of sufficient depth, and the other as a shallow, yet wide branch consisting of only a few convolutions. As a result, two-branch networks preserve partial information that is often lost after downsampling operations [57], [12], [58].

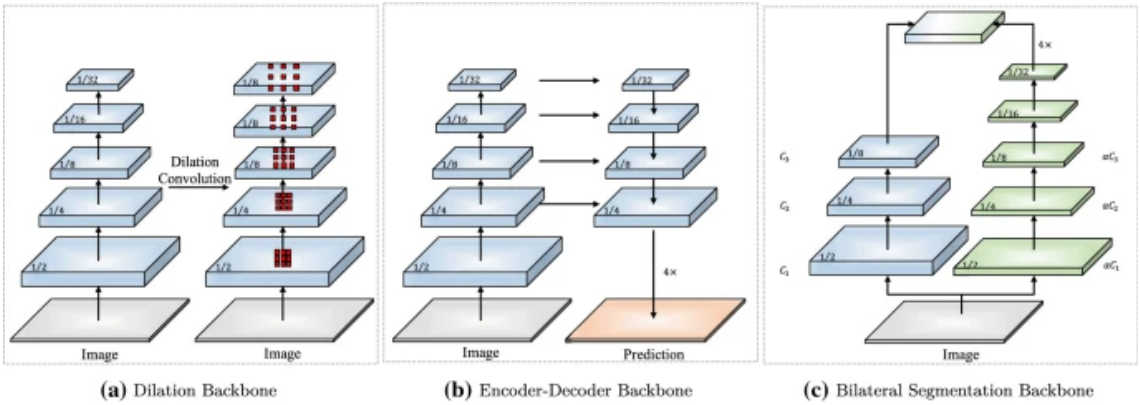

Image 5.4: *Illustration of different backbone architectures. a is the dilation backbone network. b is the encoder-decoder backbone network. c bilateral segmentation backbone network. (adapted from [12])*

5.8 Block-Based Processing with Convolutional Neural Networks

To reduce inference time, block-based processing is another effective technique. In this method, the image is divided into blocks and based on the importance of each block, its resolution is downscaled, leading to a reduction in computational costs and memory usage. For instance, Segblocks [59] employs this technique for image segmentation.

5.9 Pruning

In order to produce faster, more accurate, and more memory-efficient models, pruning can be utilized. Pruning is a method where the network tries to create more efficient representations thus reducing the number of connections and nodes used. The idea stems from the fact that visual information is highly spatially redundant, and thus can be compressed into a more efficient representation. There are two types of pruning: weight pruning and channel pruning.

Weight pruning is a method that removes unnecessary connections (parameters) in a neural network, resulting in a sparse model that still retains the high-dimensional features of the original network. Researchers [13] proposed a three-step method for

weight pruning: (1) training the network to identify important connections, (2) pruning unessential connections, and (3) fine-tuning remaining connections.

Filter pruning is a method that reduces computation costs by removing filters and their corresponding feature maps that have little effect on accuracy [60]. He et al. [61] proposed a channel-pruning approach that improves inference time while preserving accuracy. However, this approach sacrifices spatial and functional information, resulting in reduced effectiveness.

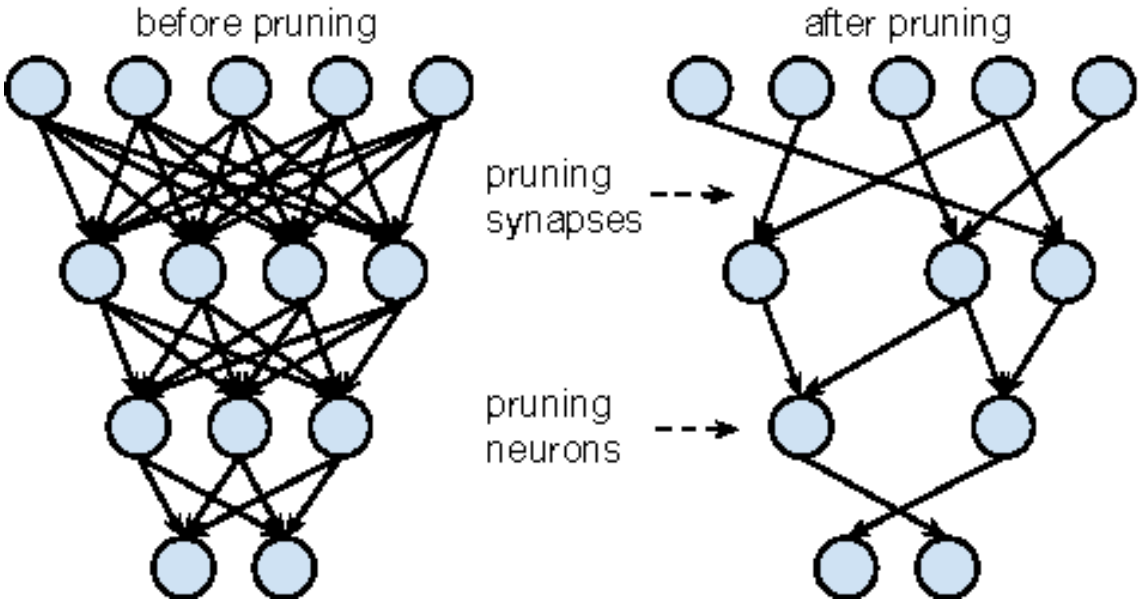

Image 5.5: *Synapses and neurons before and after pruning. (adapted from [13])*

5.10 Quantization

Using 32 bits to represent network weights can adversely affect network efficiency due to the high computational cost and memory requirements associated with 32-bit operations. In a study by Takos et al. [62], it was shown that reducing the number of bits used to represent each connection from 32 to 5 can significantly reduce computational costs. Han et al. [63] proposed a quantization approach that achieves this by sharing the same weights between multiple connections, effectively reducing the number of effective weights. These weights are then fine-tuned to optimize performance.

5.11 State of the art Models

5.11.1 Convolutional models

SqueezeNet

SqueezeNet is a deep learning model that focuses on image classification tasks and reduces the model size and computational requirements. It was designed to address the challenges of deploying large neural networks on resource-constrained devices like mobile phones or embedded systems. The model achieves compression by using a combination

of innovative architecture, including network-in-network structures, to reduce the number of parameters, and the "fire module" design that efficiently captures and processes features.

The primary goal of SqueezeNet is to balance model size and performance, making it more suitable for real-time applications on devices with limited computational resources. Despite its compact architecture, SqueezeNet has demonstrated competitive performance compared to larger models on image classification benchmarks, thus showcasing its effectiveness in the realm of efficient deep learning.

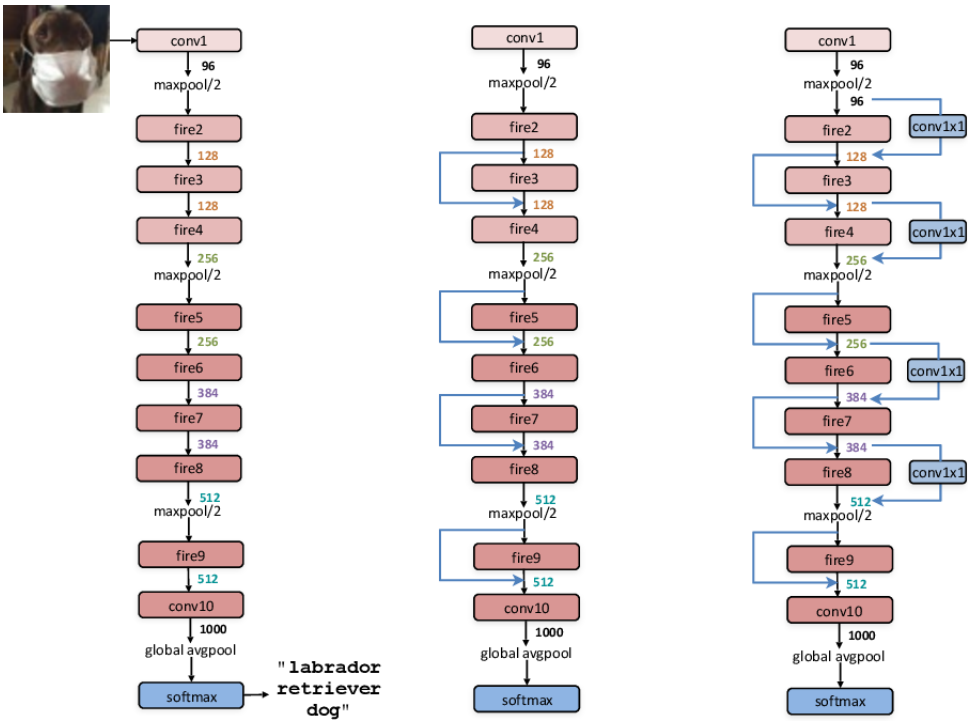

Image 5.6: *SqueezeNet architecture [14]*

FarSee-Net

FarSee-Net is a state-of-the-art approach to real-time semantic segmentation that balances high accuracy with computational efficiency. Designed as an encoder-decoder architecture, FarSee-Net introduces two key innovations: an advanced context aggregation module and a novel upsampling technique. By processing lower-resolution input data, aggregating context effectively, and restoring the high-resolution output, the network achieves an optimal trade-off between performance and resource utilization.

Zhang et al. [15] introduced the Factorized Atrous Spatial Pyramid Pooling (FASPP) module, which extends the widely adopted Atrous Spatial Pyramid Pooling (ASPP) by Chen et al. [64]. The FASPP module utilizes atrous convolutions to increase the receptive field of the filters without adding computational complexity or increasing parameters. This is accomplished by inserting r-1 zeros between adjacent kernel elements, where r is the atrous rate. While the ASPP is effective for capturing contextual information, it operates on high-dimensional feature maps, leading to increased computational cost.

To mitigate this overhead, FASPP factorizes the 3x3 atrous convolution into two stages:

- **Point-wise convolution (1x1)**: This layer linearly combines the input channels, reducing the output dimensionality and allowing for efficient channel-wise interaction.
- **Depth-wise atrous convolution (3x3)**: This layer preserves spatial context aggregation with the same kernel size and atrous rate as the original ASPP but reduces the computational burden.

By factorizing the convolutions, FarSee-Net reduces the complexity of the ASPP module, achieving faster and more efficient segmentation without compromising accuracy. To further enhance multiscale context aggregation, the network employs a cascaded version of the FASPP module, referred to as Cascaded Factorized Atrous Spatial Pyramid Pooling (CF-ASPP), which involves applying two factorized ASPP modules in sequence.

Additionally, FarSee-Net addresses the challenge of upsampling low-resolution feature maps by framing it as a super-resolution task in the feature space. Instead of conventional bilinear interpolation, which often struggles to recover fine-grained details, FarSee-Net adopts sub-pixel convolution—a technique popularized in image super-resolution tasks. During training, the network receives downsampled input images while the high-resolution label maps serve as ground truth. In the decoder, sub-pixel convolution gradually upscales the feature maps by rearranging their elements through a periodic shuffling operation, significantly improving the network's ability to recover high-resolution details. Compared to traditional deconvolution, this approach offers superior representation power and enhances the quality of the segmentation output.

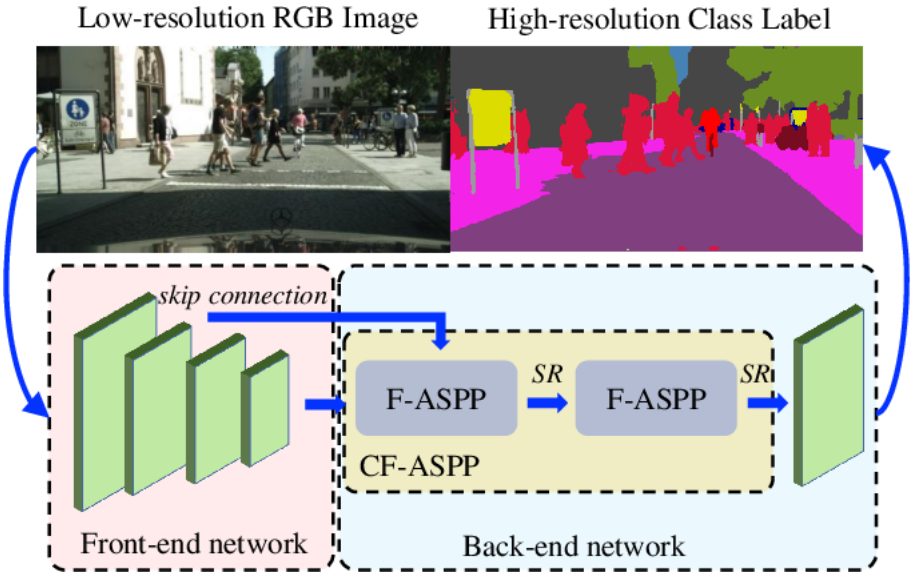

Image 5.7: *FarSee-Net architecture [15]*

5.11.2 Vision Transformer models

Segmentation Transformer (SETR)

Semantic segmentation typically involves using FCNs in an encoder-decoder architecture. The encoder is responsible for learning feature representations, while the decoder

performs pixel-level classification of these features. However, Zheng et al. [16] have proposed a new approach using a pure transformer in place of the computationally expensive stacked convolution layers-based encoder. This results in a new segmentation model called SETR.

The SETR method follows a unique approach for processing input images. It divides the image into fixed-sized patches, which are then represented using learned embeddings. These embeddings are transformed using global self-attention modeling, which helps in learning discriminative feature representations. To achieve this, a linear embedding layer is applied to the flattened pixel vectors of each patch, which results in a sequence of feature embedding vectors. These vectors are then fed as input to a transformer. The encoder transformer learns the features that are subsequently used by a decoder to reconstruct the original image resolution. Crucially, there is no downsampling in spatial resolution, but global context modeling occurs at every layer of the encoder transformer. This approach offers a completely new perspective to the semantic segmentation problem.

STETR is classified into a few variants. depending on the decoder of the model: SETR-PUP (5.8b) which has a progressive up-sampling design and the SETR-MLA (5.8c) which has a multi-level feature aggregation.

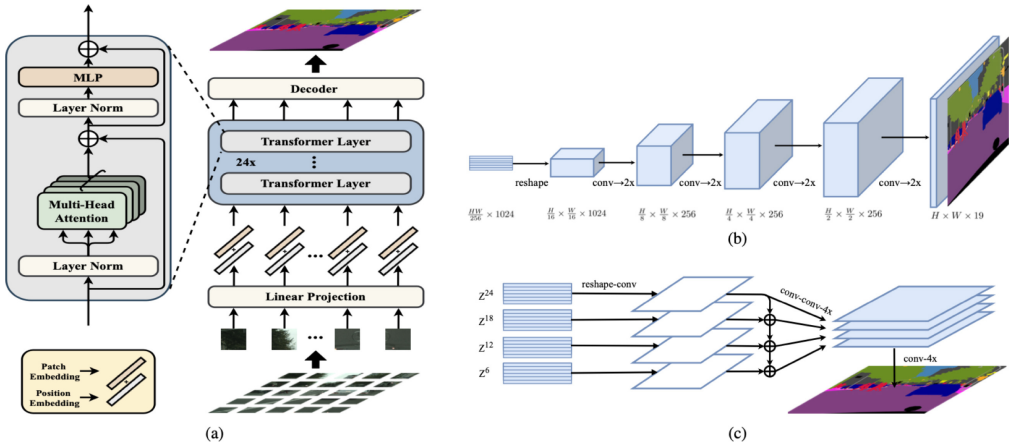

Image 5.8: *SETR architecture and its variants. (a) SETR consists of a standard Transformer. (b) SETR-PUP with a progressive up-sampling design. (c) SETR-MLA with a multi-level feature aggregation [16].*

Swin transformer

The Swin Transformer [17] is a hierarchical vision transformer model designed for dense prediction tasks such as image segmentation and object detection. Unlike traditional transformers that process the entire image at once, the Swin Transformer introduces a "shifted window mechanism" to enable more efficient computation and capture fine-grained details.

The architecture can be summarized as follows:

- **Patch Splitting and Embedding:** The input image is split into non-overlapping patches, where each patch is treated as a token. These tokens are linearly embedded to form the initial sequence of embeddings.

- **Hierarchical Structure:** The Swin Transformer operates in a hierarchical manner, where the resolution of feature maps is progressively reduced as the network deepens. This structure enables multi-scale representation learning, improving the model's ability to capture both local and global information.
- **Shifted Window Attention:** In each stage, the image is partitioned into fixed-size windows, and a "window-based multi-head self-attention" (W-MSA) mechanism is applied within each window. To enhance connections between windows, the "shifted window mechanism" shifts the window partitioning by a predefined number of pixels in alternating layers, enabling cross-window interactions without excessive computation.
- **Patch Merging:** As the network deepens, "patch merging" layers are used to reduce the number of tokens, effectively downsampling the feature maps while increasing the channel dimensions.
- **Efficient Attention:** Swin Transformer efficiently computes attention within local windows, significantly reducing the quadratic complexity of traditional attention mechanisms to linear complexity with respect to image size.

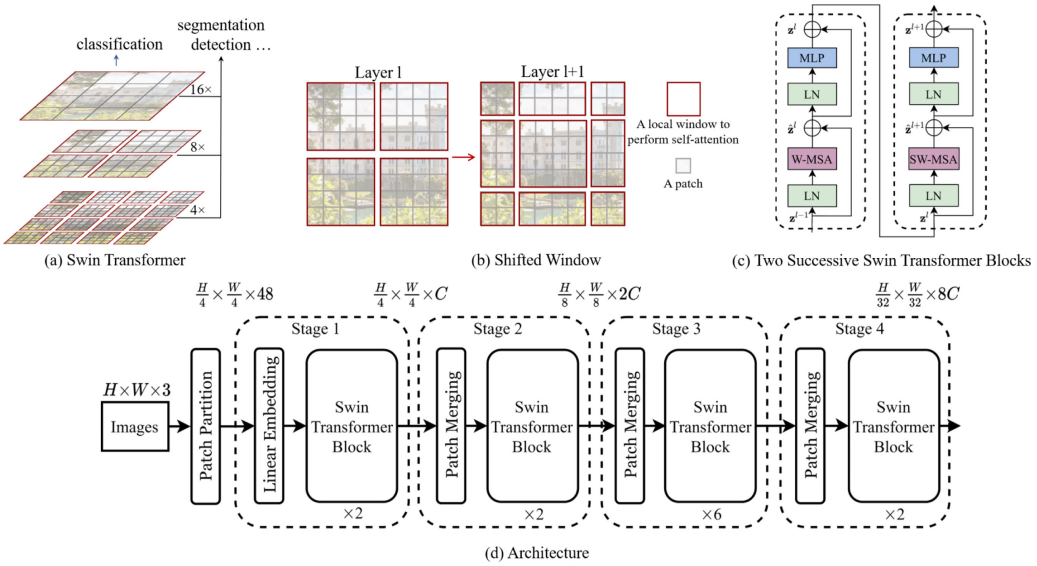

Image 5.9: *An overview of the Swin Transformer. (a) Hierarchical feature maps for reducing computational complexity. (b) Shifted window approach which was used when calculating self-attention. (c) Two successive Swin Transformer Blocks which presented at each stage. (d) The core architecture of the Swin. [17]*

Segmenter

Segmenter [18] is a transformer-based architecture designed for semantic segmentation, leveraging the strengths of self-attention mechanisms to capture long-range dependencies in image data. Unlike traditional CNN-based architectures, Segmenter employs Vision Transformers (ViT) as the backbone for feature extraction. The input image is first divided into fixed-size patches, which are then linearly embedded into a sequence of tokens.

The architecture consists of two main components:

- **Encoder**: The Vision Transformer (ViT) encoder processes the sequence of tokens through multiple transformer layers, allowing for global context modeling at each stage. Self-attention mechanisms enable the model to capture both local and global relationships between pixels.

- **Decoder**: The decoder is responsible for transforming the tokenized output of the encoder back into a high-resolution segmentation map. Segmenter uses a mask-based approach where the class tokens output from the transformer are decoded into segmentation masks. This process efficiently recovers spatial details while preserving the contextual information learned by the transformer.

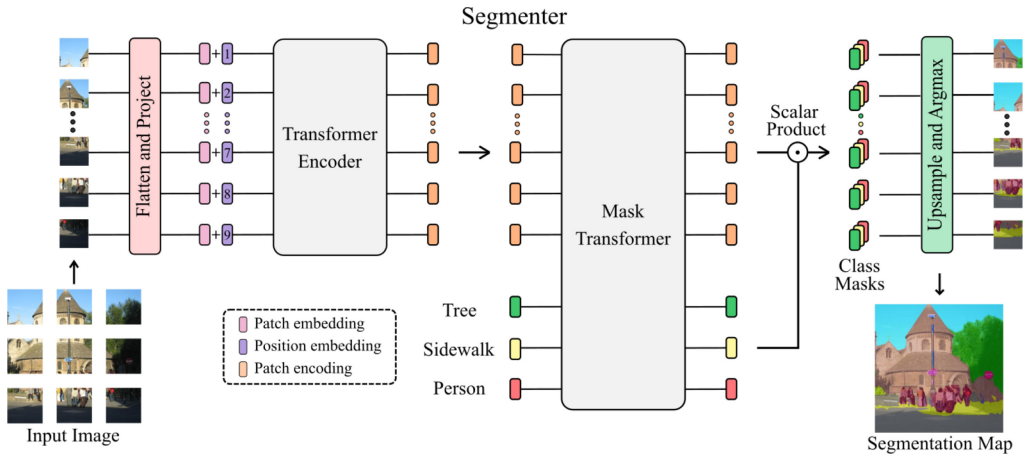

Image 5.10: *Segmenter architecture. It basically has a ViT backbone with a mask transformer as the decoder. [18]*

Masked-attention mask transformer (Mask2Former)

Mask2Former [19] is a cutting-edge transformer-based architecture designed for various segmentation tasks, including instance, panoptic, and semantic segmentation. It has demonstrated superior performance compared to existing state-of-the-art architectures, thanks to its innovative use of masked attention and a carefully designed transformer decoder.

At the core of Mask2Former is its "transformer decoder with masked attention". In contrast to traditional transformers, where attention is applied across the entire feature map, Mask2Former employs a "masked attention" mechanism that restricts the cross-attention to the predicted mask region. By limiting attention to the foreground region, the model significantly improves its computational efficiency and focus, allowing more precise predictions within the areas of interest.

The architecture is composed of three main components and is showcased in image 5.11:

- **Backbone Feature Extractor:** The backbone is responsible for generating multi-scale feature maps from the input image. Mask2Former is flexible in its choice of

backbone, supporting both convolutional neural networks (CNNs) and transformer-based models. The flexibility in backbone selection allows the architecture to benefit from different types of feature extraction techniques, depending on the specific use case.

- **Pixel Decoder:** The pixel decoder in Mask2Former builds on advancements over its predecessor, MaskFormer [65]. Specifically, it incorporates a "multi-scale deformable attention Transformer" (MSDeformAttn) [66], which enables efficient multi-scale feature aggregation. MSDeformAttn adaptively focuses on relevant regions across different scales, improving the model's ability to capture both fine details and larger contextual information.
- **Transformer Decoder:** The transformer decoder is enhanced with "masked attention", which plays a critical role in refining segmentation masks. This mechanism applies attention only to the predicted mask regions, instead of the entire feature map, thereby reducing redundant computations and focusing the model's capacity on refining mask boundaries. As a result, the model not only becomes more efficient but also produces higher-quality segmentation outputs.

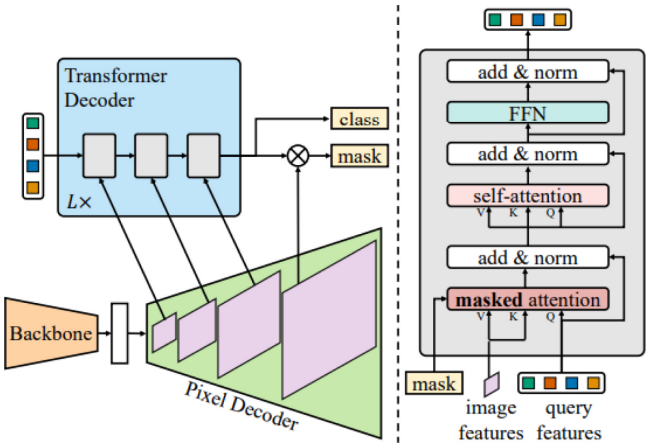

Image 5.11: *Mask2Former architecture. The model consists of a backbone feature extractor, a pixel decoder, and a Transformer decoder [19].*

Despite being a universal segmentation architecture, Mask2Former still requires task-specific training. This limitation is common among universal models, which, despite their flexibility across tasks, necessitate specialized training for each type of segmentation (e.g., instance, panoptic, or semantic).

SegFormer

Segformer [20] is a novel architecture designed for efficient semantic segmentation, combining the strengths of both transformers and convolutional neural networks (CNNs). It is characterized by its ability to deliver high performance while maintaining a lightweight structure, making it suitable for real-time applications.

The Segformer architecture consists of two main components:

- **Backbone Network:** Segformer employs a hierarchical transformer-based backbone that effectively captures multi-scale features from the input image. This backbone uses a **Mixing Transformer (MiT)** [20] structure that allows for efficient feature extraction and representation across various scales. Each layer in the MiT structure integrates information from different spatial resolutions, enabling the model to handle both fine details and broader contextual information.

- **Segmentation Head:** The segmentation head is designed to produce accurate semantic segmentation maps from the feature representations generated by the backbone. It utilizes a lightweight and flexible decoder that aggregates features from different levels of the backbone, ensuring rich contextual information is retained. This design enables Segformer to achieve high-quality segmentation results without excessive computational overhead.

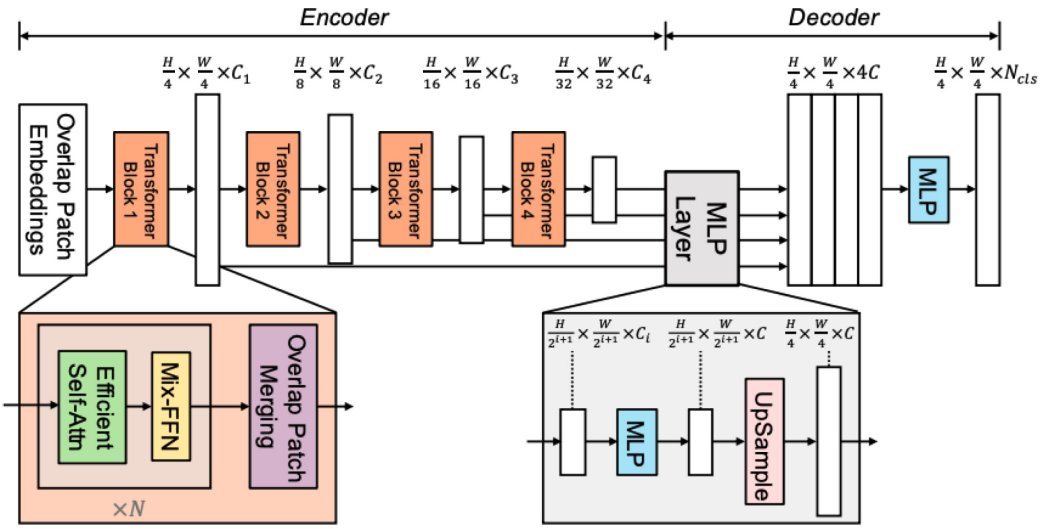

Image 5.12: *SegFormer architecture. It has a hierarchical Transformer encoder for feature extraction and a lightweight MLP decoder for predicting the final mask [20].*

One of the key innovations of Segformer is its **efficient transformer design**, which minimizes the computational burden typically associated with traditional transformer architectures. By leveraging both global and local attention mechanisms, Segformer strikes a balance between computational efficiency and performance, making it competitive with existing state-of-the-art methods while being easier to deploy in resource-constrained environments.

DAFormer

DAFormer [21] is a cutting-edge architecture specifically designed for semantic segmentation tasks, emphasizing data efficiency and robust performance. It introduces a unique approach that leverages the strengths of both convolutional neural networks (CNNs) and transformers to effectively capture contextual information while maintaining computational efficiency.

The key components of DAFormer are as follows:

- **Mix Transformers (MiT) as encoder** : Since robustness is an important property in order to achieve good domain adaptation performance as it fosters the learning of domain-invariant features, transformers are a good choice for domain adaptation as they fulfill these criteria. The encoder follows the design of MiTs [20], which are tailored for semantic segmentation. (5.11.2)

- **Context Aggregation Module (CAM):** DAFormer employs a context aggregation module that enhances the model's ability to capture long-range dependencies and contextual information. The decoder utilizes not only the context information of the bottleneck features but the context across features from different encoder levels as well. They provide valuable low-level concepts for semantic segmentation at a high resolution, which can also provide important context information.

- **Efficient Feature Fusion (lightweight design):**

  Before the feature fusion, the feature map of each level is embedded to the same number of channels by a 1×1 convolution and then are bilinearly upsampled to the size of F1, and concatenated. Multiple parallel 3×3 depthwise separable convolutions with different dilation rates are used for the context-aware feature fusion in a similar fashion to ASPP [64](5.11.1).

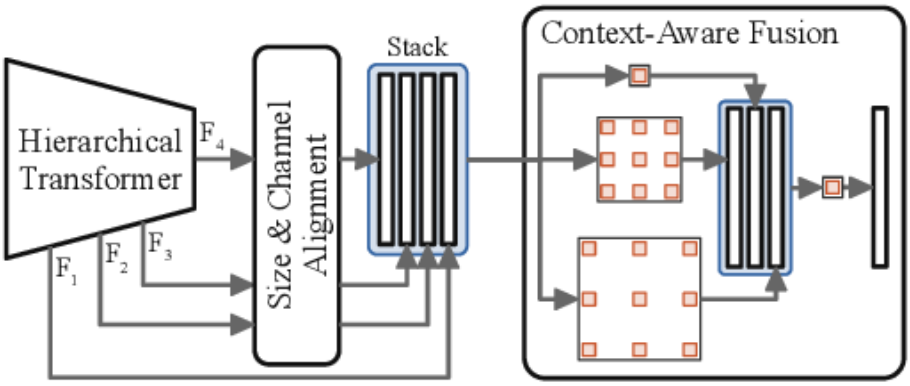

Image 5.13: *DAFormer network [21]*

Chapter 6

Datasets

6.1 Autonomous Driving Datasets

In this section, we present some common datasets used for semantic segmentation models. The choice of a suitable dataset is of great importance for the training and evaluation of the created models. The challenging task of dataset selection is one of the first major steps in research, especially for a difficult and demanding scientific field, such as autonomous driving, in which the vehicle exposure environment can be complex and varied. The datasets are divided into two categories: real and synthetic datasets.

6.1.1 Real Datasets

Cityscapes Dataset

The Cityscapes dataset [22] is a large-scale benchmark dataset specifically designed for urban scene understanding, with a focus on semantic segmentation. It consists of high-resolution images of street scenes collected from 50 cities across Germany and neighboring countries, captured under various weather conditions, seasons, and times of day.

The dataset contains:

- **Images:** A total of 5,000 finely annotated images, divided into 2,975 images for training, 500 for validation, and 1,525 for testing. Additionally, there are 20,000 coarsely annotated images for further pre-training or training.
- **Annotations:** Each image is labeled with 30 visual classes, of which 19 are used for semantic segmentation. These classes include road, sidewalk, building, vegetation, car, person, bicycle, and more. The annotations focus on pixel-level precision to provide high-quality labels.
- **Resolution:** The images have a resolution of 2048x1024 pixels, making Cityscapes a high-resolution dataset suitable for detailed segmentation tasks.
- **Tasks:** The dataset supports a variety of tasks including pixel-level semantic segmentation, instance segmentation, and panoptic segmentation, making it a versatile benchmark for evaluating model performance on urban scene understanding.

Cityscapes is widely used as a benchmark for semantic segmentation and other vision tasks in complex urban environments. Its challenging, diverse set of scenes, high-quality annotations, and high resolution have made it a standard in the field for evaluating the performance of segmentation models.

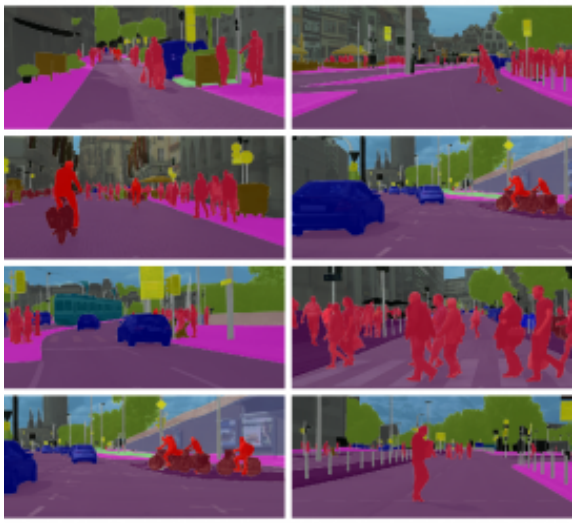

Image 6.1: *Cityscapes Dataset [22]*

Mapillary Vistas Dataset

The Mapillary Vistas dataset [23] is a large-scale, richly annotated street-level imagery dataset designed for scene understanding tasks such as semantic segmentation. Collected from diverse environments worldwide, it represents a broad variety of scenes, weather conditions, and perspectives.

The key characteristics of the Mapillary Vistas dataset include:

- **Images:** The dataset consists of over 25,000 high-resolution images sourced from cities and rural areas across different continents, ensuring a wide variety of environments, lighting conditions, and camera perspectives.
- **Annotations:** The dataset offers detailed pixel-level annotations for 124 object categories. These categories cover a wide range of semantic classes, including roads, buildings, vehicles, pedestrians, traffic signs, vegetation, and more. The precise labeling of small objects and fine details ensures high-quality training data.
- **Resolution:** Images are high resolution, varying between 1920x1080 and 4000x6000 pixels, making it suitable for fine-grained segmentation and detailed scene understanding tasks.
- **Tasks:** Mapillary Vistas is designed primarily for pixel-level semantic segmentation, with its detailed annotations supporting training and evaluation of models in complex environments. It also provides potential for other scene understanding tasks like instance and panoptic segmentation.

- **Diversity:** One of the dataset's strengths is its diversity, featuring images from urban and rural environments, captured under various weather and lighting conditions. This diversity helps models generalize better across different domains and real-world settings.

The Mapillary Vistas dataset is considered a robust and challenging benchmark for modern segmentation models, offering comprehensive annotations across diverse environments and a wide range of object classes.

Image 6.2: *Mapillary Vistas Dataset [23]*

Adverse Conditions Dataset with Correspondences

The Adverse Conditions Dataset with Correspondences (ACDC) [24] is a dataset specifically designed for semantic segmentation in challenging visual conditions. It focuses on scenes captured in adverse weather and lighting conditions, providing a robust benchmark for training and evaluating models that need to perform well in suboptimal environments.

Key characteristics of the ACDC dataset include:

- **Images:** The dataset consists of 4,006 high-resolution images collected from urban driving scenes in various European cities.
- **Adverse Conditions:** The dataset is divided into four challenging conditions: *Fog*, *Night*, *Rain*, and *Snow*, allowing researchers to evaluate model performance in visually difficult scenarios.
- **Annotations:** ACDC provides pixel-level annotations for 19 semantic classes, which are consistent with the Cityscapes label set, making it compatible with models trained on other urban scene datasets. The annotations cover a wide range of object categories, including roads, vehicles, pedestrians, buildings, vegetation, and other urban elements.
- **Resolution:** The images are high-resolution (1920x1080 pixels), suitable for capturing fine details that are essential in difficult conditions like fog or low light.
- **Split:** The dataset is divided into training, validation, and test sets. Each condition contains a balanced number of images, ensuring fair evaluation across different adverse weather scenarios.

- **Purpose:** ACDC is designed to push the boundaries of current semantic segmentation models by testing their robustness in adverse conditions, where visual features might be obscured by environmental factors like snow or darkness. This makes it a valuable resource for developing autonomous driving systems that need to operate reliably in all weather and lighting conditions.

The ACDC dataset is a valuable benchmark for improving the robustness of segmentation models in real-world, challenging conditions, making it essential for advancing the performance of autonomous driving systems.

(a) Input image I (b) Stage 1 annotation (draft) (c) Corresponding image I' (d) Stage 2 annotation (GT) (e) Invalid mask J

Image 6.3: *ACDC Dataset [24]*

6.1.2 Synthetic Datasets

Synthia Dataset

The Synthia dataset [25] is a synthetic dataset designed for training and evaluating semantic segmentation models, particularly for autonomous driving tasks. It provides a wide variety of urban scenarios with pixel-level annotations, offering a valuable resource for both training and benchmarking in environments that mimic real-world conditions.

Key characteristics of the Synthia dataset include:

- **Synthetic Data:** Synthia is a fully synthetic dataset, generated using a realistic 3D engine to simulate urban driving environments. This approach allows for the creation of diverse and highly controlled scenarios, which can be difficult or expensive to collect in the real world.
- **Scenarios:** The dataset covers a range of driving scenarios, including different weather conditions, seasons, lighting conditions (day/night), and various urban layouts like highways, residential areas, and city centers.
- **Annotations:** Synthia provides dense pixel-level annotations for up to 13-16 classes in most sequences, such as roads, sidewalks, pedestrians, vehicles, traffic signs, and more. It also supports other tasks like depth estimation and optical flow.
- **Resolution and Perspectives:** The images are available in high resolution (960x720 pixels), and the dataset includes various camera perspectives (front, left, right, rear) to simulate the full 360-degree view typically needed for autonomous driving systems.
- **Purpose:** Due to its synthetic nature, Synthia is particularly useful for tasks like domain adaptation, where models trained on synthetic data are later fine-tuned or evaluated on real-world datasets like Cityscapes or Mapillary Vistas. The diversity of

environmental conditions also helps in creating models that generalize well across different driving situations.

- **Split:** The dataset is split into various sequences that simulate continuous driving in different environments, offering over 200,000 annotated frames.

The Synthia dataset is a valuable resource for advancing research in semantic segmentation, particularly for autonomous driving applications, as it offers controlled and diverse data that complements real-world datasets.

Image 6.4: *SYNTHIA Dataset [25]*

GTA5 Dataset

The GTA5 dataset [26] is a large-scale synthetic dataset widely used for semantic segmentation, specifically for domain adaptation tasks. It is generated using the Grand Theft Auto V video game engine and provides dense pixel-level annotations for urban street scenes. The dataset is designed to resemble real-world driving scenarios, closely matching datasets like Cityscapes in terms of scene layout and labeling structure.

- The dataset contains 24,966 images rendered at a resolution of 1914x1052 pixels, covering a wide range of weather conditions, lighting variations, and urban environments.
- It includes pixel-level annotations for 19 classes, which align with those in the Cityscapes dataset, making it a popular choice for synthetic-to-real domain adaptation.
- Each image in the dataset is labeled with classes such as road, sidewalk, building, traffic light, and pedestrian, simulating real-world urban driving conditions.
- The synthetic nature of the dataset allows for efficient data collection and annotation, offering a cost-effective solution for training deep learning models in autonomous driving and segmentation tasks.

The GTA5 dataset has been instrumental in advancing research on domain adaptation, enabling models trained on synthetic data to generalize effectively to real-world scenarios.

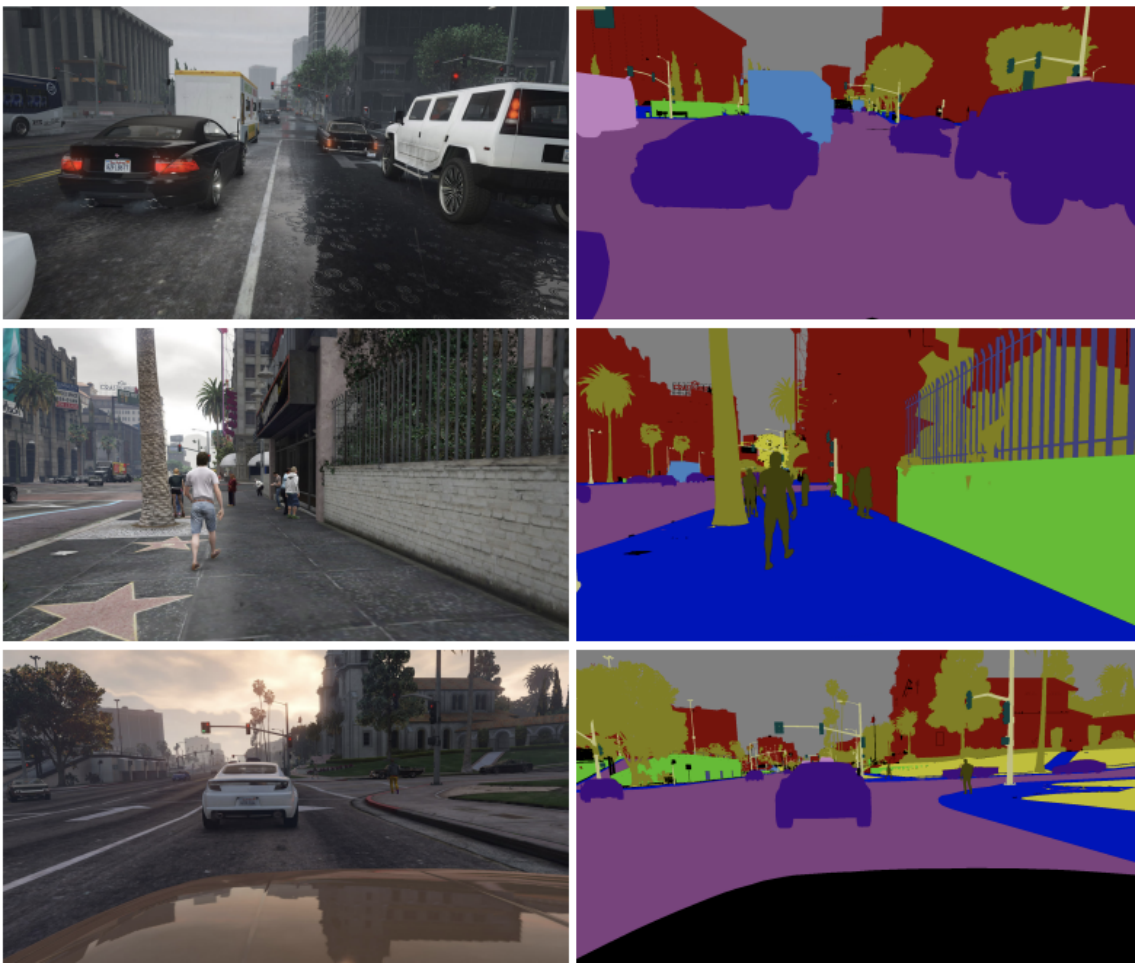

Image 6.5: *GTA5 Dataset [26]*

6.2 Other fields of knowledge

This section explores the datasets, used for other fields of knowledge, that will be utilized in the experiments of chapter 7. Three datasets will be examined: the UAVID dataset [27], the Medical Decathlon Prostate dataset, and the ACDC cardiac segmentation challenge dataset [67].

6.2.1 Uavid Dataset

The UAVID dataset is a large-scale dataset designed specifically for urban scene semantic segmentation using aerial imagery captured by Unmanned Aerial Vehicles (UAVs). It consists of high-resolution images (4096 $\times$ 2160 pixels) captured from various urban scenes across multiple cities. The dataset includes 42 sequences with over 4200 labeled images, annotated with 8 different semantic categories such as buildings, roads, trees, and cars. UAVID aims to advance research in aerial segmentation and improve the robustness of models for urban scene understanding from a bird ' s-eye view.

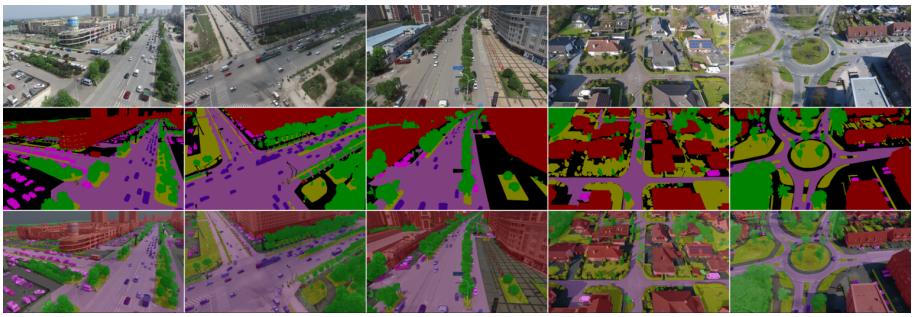

Image 6.6: *Uavid Dataset [27]*

6.2.2 Medical Decathlon Prostate dataset

The Medical Decathlon Prostate dataset includes a total of 148 patients and is composed of the following sources:

- **NCI-ISBI-2013:** Two datasets from the 2013 NCI-ISBI competition, with images acquired from both 1.5T and 3T MRI scanners from different institutions [68], [69], [70]. These datasets are labeled as A and B in the results.
- **I2CVB:** A dataset from the Initiative for Collaborative Computer Vision Benchmarking, acquired using a 3T Siemens MRI scanner with multiple imaging techniques (T2-W, DCE, DWI, MRSI) [68], [69], [71]. It is labeled as C in the results.
- **PROMISE12:** Three datasets from the PROMISE12 competition, collected from different medical centers with varying acquisition methods [68], [69], [72]. These are labeled as D, E, and F.
- **Medical Decathlon Dataset:** A new dataset provided by the Medical Decathlon Challenge, used for training and validation in within-distribution experiments [73]. It is labeled as G.

6.2.3 ACDC cardiac segmentation challenge dataset

The ACDC cardiac segmentation challenge dataset [67] is a dataset from the cardiac segmentation challenge. This dataset consists of 100 MRI scans, with annotations for the left ventricle, myocardium, and right ventricle. The slices used correspond to end-diastole and end-systole periods. We evaluate the following types of corruption created by the TorchIO software [74]:

- **Motion:** Simulates random motion artifacts caused by physiological organ movement during MRI acquisition.
- **Spike:** Generates random spike artifacts, also known as Herringbone artifacts, causing stripes across the image due to electromagnetic field spikes.
- **Ghosting:** Introduces random ghosting artifacts, usually caused by cardiac or patient motion during the scan or blood flow.

- **Bias Field:** Simulates random intensity fluctuations due to MRI field inhomogeneities.

Part II

Experiments

Chapter 7

Experimental Implementation

In this chapter, the experiments will be presented and analyzed. The datasets and the metrics will be showcased as well. Finally, we will list the models that will be tested and we will showcase the setup used for the experiments.

7.1 Evaluation Metrics

Mean Intersection over Union (mIoU)

The mean Intersection over Union (mIoU) is a widely used evaluation metric in semantic segmentation tasks. It measures the overlap between predicted and ground truth segmentation masks, making it a robust indicator of the model's performance across different classes.

The mIoU metric is computed by taking the average Intersection over Union (IoU) across all classes. The IoU for a single class is defined as the ratio between the intersection and the union of the predicted segmentation and the ground truth segmentation for that class. For a given class c, the IoU is calculated as:

$$\text{IoU}_c = \frac{|P_c \cap G_c|}{|P_c \cup G_c|}$$

Where P_c is the set of pixels predicted to belong to class c, G_c is the set of ground truth pixels for class c, $|P_c \cup G_c|$ is the total number of pixels that are predicted or ground truth for class c, including true positives, false positives, and false negatives. Thus, the IoU for a given class quantifies the overlap between the predicted and actual regions for that class, normalized by the total region. The mIoU is the mean of the IoUs over all C classes and is computed as:

$$\text{mIoU} = \frac{1}{C}\sum_{c=1}^{C} \text{IoU}_c = \frac{1}{C}\sum_{c=1}^{C} \frac{|P_c \cap G_c|}{|P_c \cup G_c|}$$

7.2 Experiments

We will conduct three experiments. The first experiment will be performed using the domain generalization pipeline (presented in section 4.3) presented in the work of [29].

The source dataset will be the GTA5 dataset(6.1.2) and the target dataset will be the Cityscapes Dataset(6.1.1). The rest of the experiments will examine how this models perform in other fields of knowledge. In particular, in the second experiment, we will train and test these models on the Uavid dataset [27], a dataset used for training models to be deployed on unmanned aerial vehicles. In the final experiments, we will be using two medical datasets: the Medical Decathlon Prostate dataset [73], and the ACDC dataset from the cardiac segmentation challenge [67]

7.2.1 Experiment 1

Configuration Details

In this experiment, we utilize the domain generalization pipeline showcased in section 4.3. The models are trained on the GTA5 dataset and evaluated on the Cityscapes dataset. The data input images are cropped to 512x512 pixel resolution for both the training and evaluation phases. The models were trained for 416.100 iterations. The metric used for the evaluation will be the mIoU metric and the optimizer used will be the Adam optimizer [75]. A batch size of 3 was used for this experiment. The setup used for this experiment consisted of an NVIDIA RTX 4000 Ada Generation Graphics Card with 20GB of video memory.

7.2.2 Experiment 2

Configuration Details

In the second experiment, the models will be trained and tested on the Uavid dataset [27]. The optimizer used will be the Adam optimizer again, the metric used will be the mIoU metric. The data input images are cropped to 1024x1024 pixel resolution for both the training and evaluation phases. The models were trained for 50 epochs. The setup used for this experiment consisted of an NVIDIA RTX 4000 Ada Generation Graphics Card with 20GB of video memory. Due to memory limitations, gradient accumulation was used in order to achieve a virtual batch size of 16 images.

7.2.3 Experiment 3

Configuration Details

In the final experiment, the models will be trained and tested on the Medical Decathlon Prostate dataset and the ACDC dataset from the cardiac segmentation challenge. The optimizer used will be the Adam optimizer again, the metric used will be the mIoU metric. The data input images are cropped to 224x224 pixel resolution for both the training and evaluation phases. The models were trained for 20 epochs. The setup used for this experiment consisted of an NVIDIA RTX 3070 Graphics Card with 8GB of video memory. Finally, this experiment's batch size will be 8 images.

7.3 Models

The models trained and evaluated are listed below:

Backbone	Decoder	Model Size(GB)	Model Params(M)	FLOPs (GFLOPs)
ResNet 18 [76]	FarSee-Net [15]	0.063	16.6	13.09
MiT-B5 [20]	Segformer [20]	0.325	85.1	110.02
MiT-B5 [20]	DaFormer [21]	0.326	85.6	126.51
MiT-B5 [20]	FarSee-Net [15]	0.328	86.2	83.51

Table 7.1. *The models tested and their sizes in terms of parameters(M) and GB.*

To assess the real-time segmentation capabilities of these models, we performed a speed test using the code available in this repository [77]. The code generates random images with a resolution of 512x512 and inputs them into the network. Subsequently, it calculates the frames per second (fps) at which the model can perform segmentation. Below are the results:

Model	FPS
FarSee-Net	**347.8**
Segformer	21.66
DaFormer	19.5
FarSee-Net2	24.5

Table 7.2. *Frames segmented per second by every model.*

The FarSee-Net model is the only one capable of achieving real-time segmentation, thanks to its lightweight encoder and decoder. Among the three transformer-based models, only FarSee-Net2, which uses the lightweight FarSee-Net decoder, is capable of achieving close to real-time speeds.

Chapter 8

Presentation and Analysis of the Results

In this chapter, the results from the experiments will be presented and analyzed.

8.1 Experiment 1

8.1.1 Results

Models	Road	S.walk	Build.	Wall	Fence	Pole	Tr.Light	Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike	mIoU
GTA5 → Cityscapes																				
FarSee-Net	73.3	26.5	75.8	20.1	2.38	22.3	12.7	3.28	79.0	30.1	81.4	31.9	10.3	60.9	11.5	13.6	0.47	8.91	5.78	30.0
Segformer	87.7	33.0	84.8	34.1	**27.4**	35.2	**47.4**	20.5	**87.8**	42.2	86.9	65.2	35.0	88.7	45.4	46.0	21.8	**29.6**	**30.2**	49.9
DAFormer	**90.0**	**45.0**	85.4	**36.4**	26.4	**37.7**	44.7	**23.0**	87.5	**42.7**	88.0	**68.5**	39.0	**89.0**	45.1	42.5	29.5	27.7	28.3	**51.4**
FarSee-Net2	88.7	34.9	**85.6**	36.1	26.5	32.4	43.2	20.9	87.1	39.0	**88.5**	65.8	**39.6**	87.3	**46.4**	**49.7**	**36.7**	26.7	27.8	50.7

Table 8.1. *Comparison of models on domain generalization pipeline. The models were trained for 416.100 iterations, using synthetic data from the GTA5 dataset and evaluated on the Cityscapes dataset. The table contains the IoU achieved by each model for each class.*

Model	Memory(GB)
FarSee-Net	**3.8**
Segformer	19.22
DaFormer	20.0
FarSee-Net2	17.4

Table 8.2. *Memory demand by every model during training.*

8.1.2 Analysis

The training and evaluation of the models spanned approximately five days for the transformer models, while the convolutional model required around two days. Notably, the convolutional model demonstrated significantly lower memory usage during training, making it less resource-intensive. However, this efficiency came at a cost: its performance lagged behind that of the transformer models, which exhibited comparable mean Intersection over Union (mIoU) scores. All classes experienced a noticeable drop in accuracy, particularly in the rarer categories such as train, bike, and rider, where the decline was especially pronounced.

8.2 Experiment 2

8.2.1 Results

Models	Building	Road	Tree	LowVeg	Moving Car	Static Car	Human	Clutter	**mIoU**
Uavid									
FarSee-Net	87.24	67.36	71.19	57.31	57.18	57.26	31.1	53.97	60.33
Segformer	92.67	77.44	78.32	70.6	**72.88**	**69.04**	43.42	64.91	71.16
DAFormer	92.41	78.87	78.34	**71.19**	72.32	68.69	**45.98**	66.08	71.73
FarSee-Net2	**92.55**	**79.88**	**79.24**	**71.19**	71.51	68.11	45.27	**67.23**	**71.87**

Table 8.3. *Comparison of models on Uavid Dataset. The models were trained for 50 epochs. The table contains the IoU achieved by each model for each class.*

Model	Memory(GB)
FarSee-Net	**1.3**
Segformer	14.7
DaFormer	15.5
FarSee-Net2	13.3

Table 8.4. *Memory demand by every model during training.*

8.2.2 Analysis

In this experiment, the transformer models exhibited similar performance levels, while the convolutional model showed a slight decline in accuracy—although this drop was less pronounced compared to previous findings. This outcome aligns with expectations, given the absence of significant domain shifts in this dataset. Notably, the memory consumption during training was consistent across the transformer models, highlighting their comparable efficiency in resource utilization.

8.3 Experiment 3

8.3.1 Results

Corruptions	RandBias				RandSpike				RandMotion				RandGhosting			
	LV	MYO	RV	mIoU	LV	MYO	RV	mIoU	LV	MYO	RV	mIoU	LV	MYO	RV	mIoU
FarSee-Net	92.0	81.8	86.0	86.6	61.3	29.8	45.4	45.5	**93.6**	**84.6**	87.8	**88.7**	92.5	**81.6**	**87.1**	**87.1**
Segformer	**92.3**	**82.2**	85.7	**86.7**	61.4	44.9	47.5	51.27	93.2	83.1	87.4	87.9	**92.6**	81.2	86.3	86.7
DAFormer	91.4	81.9	**86.1**	86.5	**65.6**	**49.6**	**56.1**	**57.1**	**93.6**	84.1	**88.3**	**88.7**	**92.6**	81.3	86.7	86.9
FarSee-Net2	89.6	80.1	83.6	84.4	61.2	44.6	50.2	52.0	92.6	82.8	87.1	87.5	91.7	80.6	85.2	85.8

Table 8.5. *Comparison table of models' evaluation on cardiac data. The results for each class for every corruption are listed in columns.*

Models	G	A	B	C	D	E	F	mIoU
FarSee-Net	**98.9**	77.1	58.1	**65.1**	**69.7**	**52.9**	64.4	64.55
Segformer	98.2	73.3	58.8	60.7	59.6	52.8	61.1	61.05
DAFormer	98.6	**79.9**	**65.6**	63.1	68.2	50.6	58.9	64.4
FarSee-Net2	98.0	79.1	61.3	64.2	67.5	51.7	**69.3**	**65.5**

Table 8.6. *Comparison table of models' evaluation on the prostate data. The models are trained on the G dataset and are evaluated separately on the A-F datasets.*

Model	Memory(GB)
FarSee-Net	**1.1**
Segformer	7.8
DaFormer	8.4
FarSee-Net2	6.5

Table 8.7. *Memory demand by every model during training.*

8.3.2 Analysis

In this experiment, the transformer and convolutional models demonstrate comparable performance, with the convolutional model often achieving the best results or coming in a close second in several cases. Notably, the convolutional model is the clear winner in terms of memory efficiency during training, requiring significantly less memory bandwidth compared to its transformer counterparts.

Part III

Epilogue

Chapter 9

Conclusions and Future Work

9.1 Conclusions

In this thesis, an analysis of the semantic segmentation challenge for autonomous driving was conducted, encompassing various aspects. Existing solutions in the literature were examined, and prominent models and datasets were presented. The approach to this task was framed through the lens of Domain Generalization, with the ultimate adoption of the state-of-the-art Domain Generalization pipeline developed in the works of [29], [32].

We decided to compare four models: FarSee-Net, Segformer, DAFormer, and FarSee-Net2. FarSee-Net is one of the best convolutional models used for real-time semantic segmentation. DAFormer was the model introduced in the works of [29], [32], while Segformer was its predecessor which introduced the MiT-B5 encoder [20]. FarSee-Net2 is a new architecture that uses the MiT-B5 backbone along with the efficient and computationally light FarSee-Net decoder. It was developed for this thesis to achieve faster inference and training times.

The 4 models were trained and tested on the same datasets (GTA5 $\rightarrow$ Cityscapes), as well as the same datasets regarding new fields of knowledge (UAV and medical imaging). The results showcased that the transformer models perform far better in real-world applications, thanks to their robustness and adaptability. The exception was the medical data, where the convolutional model remained competitive, assuming due to the domain shift between the training and testing data not being as prominent. FarSee-Net2 outperformed the other models on the UAV dataset and did comparably well to the best models in each of the other 2 experiments. The advantage of FarSee-Net2 lies in its lesser computational demand during deployment for testing and training.

9.2 Future Work

While transformers are robust and insusceptible to domain shifts, they are far from being a viable solution in real-time tasks. In our case, by using an efficient decoder we managed to speed up inference speed without sacrificing accuracy. However, this slight decrease in inference time is not enough to achieve real-time segmentation or speeds comparable to those of convolutional models. We suggest that an effort should be made to reduce the computational burden inflicted by transformer models, as it seems to be the

primary source of the spike in inference times when compared to convolutional architectures.

Bibliography

[1] Amy Bearman, Olga Russakovsky, Vittorio Ferrari και Li Fei-Fei. *What's the point: Semantic segmentation with point supervision*. *European conference on computer vision*, σελίδες 549–565. Springer, 2016.

[2] Sumit Saha. *A comprehensive guide to convolutional neural networks—the ELI5 way*. *Towards data science*, 15:15, 2018.

[3] *Introduction to Convolutional Neural Networks CNNs*.

[4] Ahmadsabry. *A Perfect guide to Understand Encoder Decoders in Depth with Visuals*. 2023.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly και others. *An image is worth 16x16 words: Transformers for image recognition at scale*. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Pedro O Pinheiro και Ronan Collobert. *From image-level to pixel-level labeling with convolutional networks*. *Proceedings of the IEEE conference on computer vision and pattern recognition*, σελίδες 1713–1721, 2015.

[7] Jifeng Dai, Kaiming He και Jian Sun. *Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation*. *Proceedings of the IEEE international conference on computer vision*, σελίδες 1635–1643, 2015.

[8] Yunsheng Li, Lu Yuan και Nuno Vasconcelos. *Bidirectional learning for domain adaptation of semantic segmentation*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 6936–6945, 2019.

[9] Yunhui Guo, Yandong Li, Rogério Schmidt Feris, Liqiang Wang και Tajana Simunic. *Depthwise Convolution is All You Need for Learning Multiple Visual Domains*. *ArXiv*, abs/1902.00927, 2019.

[10] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin και Jian Sun. *ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices*. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 6848–6856, 2017.

[11] Vijay Badrinarayanan, Alex Kendall και Roberto Cipolla. *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:2481–2495, 2015.

[12] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen και Nong Sang. *BiSeNet V2: Bilateral Network with Guided Aggregation for Real-Time Semantic Segmentation*. *International Journal of Computer Vision*, 129:3051 – 3068, 2020.

[13] Song Han, Jeff Pool, John Tran και William J. Dally. *Learning both Weights and Connections for Efficient Neural Network*. *Neural Information Processing Systems*, 2015.

[14] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally και Kurt Keutzer. *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size*. *arXiv preprint arXiv:1602.07360*, 2016.

[15] Zhanpeng Zhang και Kaipeng Zhang. *Farsee-net: Real-time semantic segmentation by efficient multi-scale context aggregation and feature space super-resolution*. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, σελίδες 8411–8417. IEEE, 2020.

[16] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr και others. *Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers*. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, σελίδες 6881–6890, 2021.

[17] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin και Baining Guo. *Swin transformer: Hierarchical vision transformer using shifted windows*. *Proceedings of the IEEE/CVF international conference on computer vision*, σελίδες 10012–10022, 2021.

[18] Robin Strudel, Ricardo Garcia, Ivan Laptev και Cordelia Schmid. *Segmenter: Transformer for semantic segmentation*. *Proceedings of the IEEE/CVF international conference on computer vision*, σελίδες 7262–7272, 2021.

[19] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov και Rohit Girdhar. *Masked-attention mask transformer for universal image segmentation*. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, σελίδες 1290–1299, 2022.

[20] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez και Ping Luo. *SegFormer: Simple and efficient design for semantic segmentation with transformers*. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[21] Lukas Hoyer, Dengxin Dai και Luc Van Gool. *Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation*. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, σελίδες 9924–9935, 2022.

[22] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth και Bernt Schiele. *The*

cityscapes dataset for semantic urban scene understanding. *Proceedings of the IEEE conference on computer vision and pattern recognition*, σελίδες 3213–3223, 2016.

[23] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo και Peter Kontschieder. *The mapillary vistas dataset for semantic understanding of street scenes*. *Proceedings of the IEEE international conference on computer vision*, σελίδες 4990–4999, 2017.

[24] Christos Sakaridis, Dengxin Dai και Luc Van Gool. *ACDC: The Adverse Conditions Dataset With Correspondences for Semantic Driving Scene Understanding*. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, σελίδες 10765–10775, 2021.

[25] Germán Ros, Laura Sellart, Joanna Materzynska, David Vázquez και Antonio M. López. *The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes*. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, σελίδες 3234–3243, 2016.

[26] Stephan R. Richter, Vibhav Vineet, Stefan Roth και Vladlen Koltun. *Playing for Data: Ground Truth from Computer Games*. *European Conference on Computer Vision (ECCV)*Bastian Leibe, Jiri Matas, Nicu Sebe και Max Welling, επιμελητές, τόμος 9906 στο *LNCS*, σελίδες 102–118. Springer International Publishing, 2016.

[27] Ye Lyu, George Vosselman, Gui Song Xia, Alper Yilmaz και Michael Ying Yang. *UAVid: A semantic segmentation dataset for UAV imagery*. *ISPRS journal of photogrammetry and remote sensing*, 165:108–119, 2020.

[28] Yujian Mo, Yan Wu, Xinneng Yang, Feilin Liu και Yujun Liao. *Review the state-of-the-art technologies of semantic segmentation based on deep learning*. *Neurocomputing*, 493:626–646, 2022.

[29] Lukas Hoyer, Dengxin Dai και Luc Van Gool. *Hrda: Context-aware high-resolution domain-adaptive semantic segmentation*. *European conference on computer vision*, σελίδες 372–391. Springer, 2022.

[30] Jia Deng, Wei Dong, Richard Socher, Li Jia Li, Kai Li και Li Fei-Fei. *ImageNet: A large-scale hierarchical image database*. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, σελίδες 248–255, 2009.

[31] Yuyang Zhao, Zhun Zhong, Na Zhao, Nicu Sebe και Gim Hee Lee. *Style-hallucinated dual consistency learning for domain generalized semantic segmentation*. *European conference on computer vision*, σελίδες 535–552. Springer, 2022.

[32] Lhoyer. *GitHub - lhoyer/HRDA: [ECCV22] Official Implementation of HRDA: Context-Aware High-Resolution Domain-Adaptive Semantic Segmentation*.

[33] Yann LeCun, Yoshua Bengio και Geoffrey Hinton. *Deep learning*. *nature*, 521(7553):436–444, 2015.

[34] Junsong Fan, Zhaoxiang Zhang, Chunfeng Song και Tieniu Tan. *Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 4283–4292, 2020.

[35] Yu Ting Chang, Qiaosong Wang, Wei Chih Hung, Robinson Piramuthu, Yi Hsuan Tsai και Ming Hsuan Yang. *Weakly-supervised semantic segmentation via sub-category exploration*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 8991–9000, 2020.

[36] Deepak Pathak, Philipp Krahenbuhl και Trevor Darrell. *Constrained convolutional neural networks for weakly supervised segmentation*. *Proceedings of the IEEE international conference on computer vision*, σελίδες 1796–1804, 2015.

[37] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein και Bernt Schiele. *Simple does it: Weakly supervised instance and semantic segmentation*. *Proceedings of the IEEE conference on computer vision and pattern recognition*, σελίδες 876–885, 2017.

[38] Wei Xia, Csaba Domokos, Jian Dong, Loong Fah Cheong και Shuicheng Yan. *Semantic segmentation without annotating segments*. *Proceedings of the IEEE international conference on computer vision*, σελίδες 2176–2183, 2013.

[39] Rui Qian, Yunchao Wei, Honghui Shi, Jiachen Li, Jiaying Liu και Thomas Huang. *Weakly supervised scene parsing with point-based distance metric learning*. *Proceedings of the AAAI Conference on Artificial Intelligence*, τόμος 33, σελίδες 8843–8850, 2019.

[40] Yanchao Yang και Stefano Soatto. *Fda: Fourier domain adaptation for semantic segmentation*. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, σελίδες 4085–4095, 2020.

[41] Zuxuan Wu, Xintong Han, Yen Liang Lin, Mustafa Gokhan Uzunbas, Tom Goldstein, Ser Nam Lim και Larry S Davis. *Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation*. *Proceedings of the European Conference on Computer Vision (ECCV)*, σελίδες 518–534, 2018.

[42] Marco Toldo, Andrea Maracani, Umberto Michieli και Pietro Zanuttigh. *Unsupervised domain adaptation in semantic segmentation: a review*. *Technologies*, 8(2):35, 2020.

[43] Xinge Zhu, Hui Zhou, Ceyuan Yang, Jianping Shi και Dahua Lin. *Penalizing top performers: Conservative loss for semantic segmentation adaptation*. *Proceedings of the European Conference on Computer Vision (ECCV)*, σελίδες 568–583, 2018.

[44] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi και Kyungnam Kim. *Image to image translation for domain adaptation*. *Proceedings of the IEEE conference on computer vision and pattern recognition*, σελίδες 4500–4509, 2018.

[45] Haoshuo Huang, Qixing Huang και Philipp Krahenbuhl. *Domain transfer through deep activation matching*. *Proceedings of the European Conference on Computer Vision (ECCV)*, σελίδες 590–605, 2018.

[46] Yi Hsuan Tsai, Wei Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming Hsuan Yang και Manmohan Chandraker. *Learning to adapt structured output space for semantic segmentation*. *Proceedings of the IEEE conference on computer vision and pattern recognition*, σελίδες 7472–7481, 2018.

[47] Matteo Biasetton, Umberto Michieli, Gianluca Agresti και Pietro Zanuttigh. *Unsupervised domain adaptation for semantic segmentation of urban scenes*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, σελίδες 0–0, 2019.

[48] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada και Kate Saenko. *Adversarial dropout regularization*. *arXiv preprint arXiv:1711.01575*, 2017.

[49] Teo Spadotto, Marco Toldo, Umberto Michieli και Pietro Zanuttigh. *Unsupervised domain adaptation with multiple domain discriminators and adaptive self-training*. *2020 25th International Conference on Pattern Recognition (ICPR)*, σελίδες 2845–2852. IEEE, 2021.

[50] Minghao Chen, Hongyang Xue και Deng Cai. *Domain adaptation for semantic segmentation with maximum squares loss*. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, σελίδες 2090–2099, 2019.

[51] Min Wang, Baoyuan Liu και Hassan Foroosh. *Factorized Convolutional Neural Networks*. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, σελίδες 545–553, 2016.

[52] Adam Paszke, Abhishek Chaurasia, Sangpil Kim και Eugenio Culurciello. *ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation*. *ArXiv*, abs/1606.02147, 2016.

[53] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens και Zbigniew Wojna. *Rethinking the Inception Architecture for Computer Vision*. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, σελίδες 2818–2826, 2015.

[54] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo και Xiaolin Wei. *Rethinking BiSeNet For Real-time Semantic Segmentation*. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, σελίδες 9711–9720, 2021.

[55] Karen Simonyan και Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. *CoRR*, abs/1409.1556, 2014.

[56] Vladimir Nekrasov, Chunhua Shen και Ian D. Reid. *Template-Based Automatic Search of Compact Semantic Segmentation Architectures*. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, σελίδες 1969–1978, 2019.

[57] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu και Nong Sang. *BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation*. *ArXiv*, abs/1808.00897, 2018.

[58] Yuanduo Hong, Huihui Pan, Weichao Sun και Yisong Jia. *Deep Dual-resolution Networks for Real-time and Accurate Semantic Segmentation of Road Scenes*. *ArXiv*, abs/2101.06085, 2021.

[59] Thomas Verelst και Tinne Tuytelaars. *SegBlocks: Block-Based Dynamic Resolution Networks for Real-Time Segmentation*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:2400–2411, 2020.

[60] Determine Filters'Importance. *Pruning Filters for Efficient ConvNets*.

[61] Yihui He, Xiangyu Zhang και Jian Sun. *Channel pruning for accelerating very deep neural networks*. *Proceedings of the IEEE international conference on computer vision*, σελίδες 1389–1397, 2017.

[62] Georgios Takos. *A Survey on Deep Learning Methods for Semantic Image Segmentation in Real-Time*. *ArXiv*, abs/2009.12942, 2020.

[63] Song Han, Huizi Mao και William J. Dally. *Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding*. *arXiv: Computer Vision and Pattern Recognition*, 2015.

[64] Liang Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy και Alan L Yuille. *Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs*. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[65] Bowen Cheng, Alex Schwing και Alexander Kirillov. *Per-pixel classification is not all you need for semantic segmentation*. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021.

[66] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang και Jifeng Dai. *Deformable detr: Deformable transformers for end-to-end object detection*. *arXiv preprint arXiv:2010.04159*, 2020.

[67] Olivier Bernard, Alain Lalande, Clement Zotti, Frederick Cervenansky, Xin Yang, Pheng Ann Heng, Irem Cetin, Karim Lekadir, Oscar Camara, Miguel Angel Gonzalez Ballester και others. *Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: is the problem solved?* *IEEE transactions on medical imaging*, 37(11):2514–2525, 2018.

[68] Nassir Navab, Joachim Hornegger, William M Wells και Alejandro Frangi. *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III*, τόμος 9351. Springer, 2015.

[69] Quande Liu, Qi Dou, Lequan Yu και Pheng Ann Heng. *MS-Net: multi-site network for improving prostate segmentation with heterogeneous MRI data*. *IEEE transactions on medical imaging*, 39(9):2713–2724, 2020.

[70] Nicholas Bloch, Anant Madabhushi, Henkjan Huisman, John Freymann, Justin Kirby, Michael Grauer, Andinet Enquobahrie, Carl Jaffe, Larry Clarke και Keyvan Farahani. *NCI-ISBI 2013 challenge: automated segmentation of prostate structures. The Cancer Imaging Archive*, 2015.

[71] Guillaume Lemaître, Robert Martí, Jordi Freixenet, Joan C Vilanova, Paul M Walker και Fabrice Meriaudeau. *Computer-aided detection and diagnosis for prostate cancer based on mono and multi-parametric MRI: a review*. *Computers in biology and medicine*, 60:8–31, 2015.

[72] PROMISE12 Challenge Organizers. *Prostate MR Image Segmentation 2012 Challenge*, 2012. Accessed: 2024-09-25.

[73] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers και others. *The medical segmentation decathlon*. *Nature communications*, 13(1):4128, 2022.

[74] Fernando Pérez-García, Rachel Sparks και Sébastien Ourselin. *TorchIO: a Python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning*. *Computer methods and programs in biomedicine*, 208:106236, 2021.

[75] P Kingma Diederik. *Adam: A method for stochastic optimization*. *(No Title)*, 2014.

[76] Kaiming He, Xiangyu Zhang, Shaoqing Ren και Jian Sun. *Deep residual learning for image recognition*. *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, σελίδες 770–778, 2016.

[77] Zh. *GitHub - zh320/realtime-semantic-segmentation-pytorch: PyTorch implementation of over 30 realtime semantic segmentations models, e.g. BiSeNetv1, BiSeNetv2, CGNet, ContextNet, DABNet, DDRNet, EDANet, ENet, ERFNet, ESPNet, ESPNetv2, FastSCNN, ICNet, LEDNet, LinkNet, PP-LiteSeg, SegNet, ShelfNet, STDC, SwiftNet, and support knowledge distillation, distributed training etc.*