National Technical University of Athens

School of Electrical and Computer Engineering

Division of Computer Science, Artificial Intelligence and Learning Systems Laboratory

Advancing UAV Safety and Efficiency through Machine Learning Based Open-Set Detection

DIPLOMA THESIS

of

Spyridon Loukovitis

Supervisor: Athanasios Voulodimos
Assistant Professor, N.T.U.A.

Athens, October 2025

Εθνικο Μετσοβιο Πολυτεχνειο
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης

Βελτίωση της Ασφάλειας και της Αποδοτικότητας των UAV με Αλγορίθμους Ανοιχτής Ανίχνευσης βασισμένους στη Μηχανική Μάθηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Σπυρίδωνος Λουκοβίτη

Επιβλέπων: Αθανάσιος Βουλόδημος
Επίκουρος Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 30 Οκτωβρίου 2025.

.....	.....	.....
Αθανάσιος Βουλόδημος	Γιώργος Στάμου	Ανδρέας-Γεώργιος Σταφυλοπάτης
Επίκουρος Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Ομότιμος Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2025

.....

Σπυρίδων Λουκοβίτης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η παρούσα διπλωματική εργασία ασχολείται με την ανάπτυξη ενός πλαισίου ανίχνευσης αντικειμένων ανοιχτού συνόλου (open-set) για εφαρμογές αέρος-αέρος με μη επανδρωμένα αεροχήματα (UAV). Η προτεινόμενη μεθοδολογία είναι ανεξάρτητη από το βασικό μοντέλο και βασίζεται στην εξαγωγή διανυσμάτων χαρακτηριστικών (embeddings) από τον ανιχνευτή, στα οποία εφαρμόζεται μοντελοποίηση εντροπίας μέσω Γκαουσιανών Μιγμάτων (Gaussian Mixture Models) για την εκτίμηση της σημασιολογικής αβεβαιότητας. Για τη βελτίωση της σταθερότητας και της διακριτικής ικανότητας, ενσωματώνονται τεχνικές φασματικής κανονικοποίησης (spectral normalization) και κλιμάκωσης θερμοκρασίας (temperature scaling), ενώ χρησιμοποιούνται στοχευμένες τεχνικές εμπλουτισμού δεδομένων με προσομοιωμένες αλλοιώσεις που αντανακλούν τις συνθήκες πτήσης. Η εργασία περιγράφει αναλυτικά τη διαδικασία ενσωμάτωσης της μεθόδου σε σύγχρονους ανιχνευτές, καθώς και την προσαρμογή της για χρήση σε ενσωματωμένα συστήματα UAV.

Λέξεις κλειδιά: Ανοιχτού Συνόλου Ανίχνευση Αντικειμένων, Μη Επανδρωμένα Οχήματα, Εκτίμηση Αβεβαιότητας, Επεξεργασία Εικόνας.

Abstract

This thesis presents the development of an open-set object detection framework for air-to-air scenarios with unmanned aerial vehicles (UAVs), aimed at enhancing perception reliability in real-world flight conditions. The proposed method is model-agnostic and operates on feature embeddings extracted from the detector, applying Gaussian Mixture Models (GMMs) to model semantic uncertainty through entropy estimation. To improve stability and discrimination, spectral normalization and temperature scaling techniques are integrated, while targeted data augmentation with simulated corruptions is employed to reflect aerial imaging conditions. The implementation process includes the adaptation of the framework for integration into modern detectors and its optimization for embedded UAV systems. The methodology is documented alongside the theoretical background, providing a practical reference for future work in UAV perception and open-set detection. This thesis was written in English to be accessible to a wider audience. A comprehensive summary in Greek follows.

Keywords: Open-Set Object Detection, Unmanned Aerial Vehicles, Uncertainty Estimation, Image Processing.

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή Αθανάσιο Βουλόδημο για την ευκαιρία που μου έδωσε να εργαστώ πάνω σε αυτή τη διπλωματική, για τους πόρους που την κατέστησαν δυνατή και για την καθοδήγησή του καθ' όλη τη διάρκειά της. Ευχαριστώ, επίσης, τον υποψήφιο διδάκτορα Βασίλειο Καραμπίνη και τον Δρ. Αναστάσιο Άρσενο για τη συνεχή τους καθοδήγηση και τη στενή συνεργασία μας, που οδήγησαν στο παρόν αποτέλεσμα, για το οποίο είμαι ιδιαίτερα περήφανος. Ιδιαίτερες ευχαριστίες οφείλω και στον καθηγητή Γεώργιο Στάμου, με τον οποίο είχα την τιμή να συνεργαστώ ακαδημαϊκά τους τελευταίους έξι μήνες.

Είμαι ευγνώμων προς όλους τους συμφοιτητές, φίλους και ανθρώπους που στάθηκαν δίπλα μου σε οποιοδήποτε σημείο της ακαδημαϊκής μου πορείας, και οι οποίοι, άμεσα ή έμμεσα, με βοήθησαν να φτάσω στο σημείο που βρίσκομαι σήμερα, περισσότερο απ' όσο ίσως οι ίδιοι συνειδητοποιούν.

Τέλος, ευχαριστώ την οικογένειά μου για τη σταθερή ψυχολογική της στήριξη σε κάθε βήμα, καθώς και για το ότι ήταν πάντα εκεί για εμένα. Ιδιαιτέρως, ευχαριστώ τον αδερφό μου, Γιώργο, για την αστείρευτη υπομονή του απέναντι στα συνεχή μου παράπονα για μαθήματα, εργασίες και πειράματα.

Λουκοβίτης Σπυρίδων
Αύγουστος 2025

Acknowledgements

First and foremost, I would like to thank my supervisor, Professor Athanasios Voulodimos, for giving me the opportunity to work on this thesis, for providing the resources that made it possible, and for his guidance throughout its duration. I would also like to thank Ph.D. candidate Vasileios Karampinis and Dr. Anastasios Arsenos for their continuous guidance and close collaboration, which led to the present outcome, of which I am particularly proud. Special thanks are also due to Professor Georgios Stamou, with whom I have had the privilege of closely collaborating academically over the past six months.

I am also grateful to all my fellow students, friends, and people who have stood by me at various points in my academic journey, and who, directly or indirectly, have helped me reach the point I am at today, more than they themselves may realize.

Finally, I would like to thank my family for their constant psychological support at every step, and for always being there for me. In particular, I would like to thank my brother, George, for his endless patience in listening to my complaints about courses, assignments, and experiments.

Loukovitis Spyridon
August 2025

Contents

List of Figures

List of Tables

Chapter 1

Εκτεταμένη περίληψη στα Ελληνικά

1.1 Εισαγωγή

Τα μη επανδρωμένα αεροσκάφη (UAVs) έχουν γνωρίσει ραγδαία εξάπλωση σε εφαρμογές όπως επιτήρηση, μεταφορές και έρευνα–διάσωση. Η αυξημένη χρήση τους ςπιφέρει αυξημένο κίνδυνο εναέριων συγκρούσεων, ειδικά σε πυκνό ή αστικό εναέριο χώρο. Για την αντιμετώπιση αυτού του προβλήματος αναπτύσσονται συστήματα Sense-and-Avoid (SAA), τα οποία επιτρέπουν στα UAVs να ανιχνεύουν και να αποφεύγουν αυτόνομα εμπόδια. Ωστόσο, οι υφιστάμενες προσεγγίσεις δυσκολεύονται με μη συνεργατικούς στόχους (π.χ. άλλα UAVs ή πουλιά) και σε δύσκολες περιβαλλοντικές συνθήκες.

Ένα κρίσιμο ζήτημα είναι ότι τα περισσότερα συστήματα ανίχνευσης λειτουργούν με την υπόθεση του κλειστού συνόλου (closed set), δηλαδή ότι όλες οι κατηγορίες αντικειμένων είναι γνωστές εκ των προτέρων. Στην πράξη όμως, τα UAV συχνά συναντούν άγνωστα αντικείμενα, με αποτέλεσμα υποβάθμιση της ακρίβειας και αύξηση του κινδύνου. Για αυτό τον λόγο απαιτούνται μέθοδοι ανοιχτού συνόλου (open-set detection), που μπορούν να αναγνωρίζουν γνωστές κατηγορίες αλλά και να ανιχνεύουν άγνωστες.

Στόχος αυτής της διπλωματικής είναι η ανάπτυξη και αξιολόγηση ενός πλαισίου ανίχνευσης ανοιχτού συνόλου, το οποίο ενσωματώνει εκτίμηση αβεβαιότητας σε σενάρια αέρος–αέρος. Η προσέγγιση αυτή βασίζεται σε μοντελοποίηση της εσωτερικής αναπαράστασης των ανιχνεύσεων στο μοντέλο (embeddings), σε Gaussian Mixture Models και σε τεχνικές κανονικοποίησης, ώστε να παρέχει αξιόπιστες προβλέψεις με μικρό υπολογιστικό κόστος σε πραγματικό χρόνο.

1.2 Σχετική Έρευνα

1.2.1 Ανίχνευση Αντικειμένων με UAV

Η σύγχρονη ανίχνευση αντικειμένων σε εναέρια δεδομένα βασίζεται κυρίως σε ταχείς *one-stage* ανιχνευτές (π.χ. οικογένεια YOLO) και σε *transformer-based* προσεγγίσεις (π.χ. DETR). Οι πρώτοι προσφέρουν ευνοϊκό λόγο ακρίβειας–ταχύτητας για ενσωματωμένες πλατφόρμες, ενώ οι δεύτεροι αξιοποιούν πετυχαίνουν μεγαλύτερη ακρίβεια αλλά συχνά με υψηλότερο υπολογιστικό κόστος. Στα εναέρια benchmarks, σύνολα όπως το AOT (air-to-air) και το FAIR1M/DOTA (air-to-ground, με περιστρεφόμενα πλαίσια) έχουν επιταχύνει την πρόοδο, αναδεικνύοντας όμως προκλήσεις: πολύ μικρούς στόχους, έντονες μεταβολές κλί-

μακας/προσανατολισμού και ανάγκη για γρήγορους υπολογισμούς σε πλατφόρμες με περιορισμούς SWaP.

1.2.2 Ανοιχτή Αναγνώριση (Open-Set Recognition)

Σε πραγματικές αποστολές, ο ανιχνευτής συναντά άγνωστα αντικείμενα ή συνθήκες εκτός κατανομής, άρα απαιτείται ικανότητα διάκρισης *γνωστών* από *αγνώστους*. Οι βασικές κατηγορίες μεθόδων περιλαμβάνουν: (α) *threshold-based* κανόνες πάνω σε βαθμούς εμπιστοσύνης/εντροπία, και (β) συνδυασμοί προβλέψεων από διαφορετικά μοντέλα ή από το ίδιο μοντέλο με ελαφρώς διαφορετική εικόνα εισόδου (ensembling/ bayes). Ο στόχος είναι αξιόπιστη απόρριψη OOD χωρίς υποβάθμιση του mAP και της ταχύτητας.

1.3 Θεωρητικό Υπόβαθρο

1.3.1 Αβεβαιότητα στη Μηχανική Μάθηση

Η ποσοτικοποίηση της αβεβαιότητας είναι κρίσιμη για συστήματα ασφαλείας. Διακρίνουμε δύο κύριους τύπους: **aleatoric** (προέρχεται από τον θόρυβο/ασάφεια των δεδομένων: φωτισμός, καιρικές συνθήκες, μερική απόκρυψη) και **epistemic** (προέρχεται από άγνοια του μοντέλου: ανεπαρκή δεδομένα, ελλιπή παραμετροποίηση, περιοχές του χώρου εισόδων που δεν έχουν παρατηρηθεί). Η πρώτη είναι συχνά μη μειώσιμη, ενώ η δεύτερη μπορεί να μειωθεί με περισσότερα ή καλύτερα δεδομένα και κατάλληλη μοντελοποίηση. Στην πράξη, η εκτίμηση αβεβαιότητας υλοποιείται με δείκτες εμπιστοσύνης/εντροπίας, ensembles, στοχαστικά περάσματα (π.χ. dropout), ή βαθμονόμηση θερμοκρασίας, με στόχο αξιόπιστες αποφάσεις απόρριψης/αποδοχής.

1.3.2 Κανονικοποίηση Φάσματος (Spectral Normalization)

Η κανονικοποίηση φάσματος ελέγχει τη *Lipschitz* σταθερά των στρωμάτων περιορίζοντας τη μέγιστη ιδιοτιμή (spectral norm) των βαρών. Πρακτικά, κάθε γραμμικός τελεστής W ανακλιμακώνεται σε $\bar{W} = W/\|W\|_2$, εξασφαλίζοντας 1-Lipschitz συμπεριφορά ανά στρώμα (με αποδοτική προσέγγιση της $\sigma_{\max}$ μέσω power iteration). Σε σύνθεση στρωμάτων με 1-Lipschitz ενεργοποιήσεις (ReLU κ.ά.), προκύπτει παγκόσμιος έλεγχος ομαλότητας. Για την ταξινόμηση, αυτό οδηγεί σε *καλώς δομημένους εμφωλευμένους χώρους* (embeddings) με σταθερή γεωμετρία, βελτιωμένη βαθμονόμηση και πιο αξιόπιστους δείκτες αβεβαιότητας.

1.3.3 Μείγματα Γκαουσιανών (Gaussian Mixture Models)

Τα Μείγματα Γκαουσιανών (GMMs) προσεγγίζουν πολύτροπες κατανομές ως κυρτό συνδυασμό K Γκαουσιανών:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k), \quad \sum_k \pi_k = 1.$$

Η εκτίμηση των παραμέτρων (π_k, μ_k, Σ_k) γίνεται συνήθως με EM. Στο πλαίσιο ανίχνευσης/αναγνώρισης, τα GMMs εφαρμόζονται στον χώρο εμφωλευμάτων ανά κλάση ώστε να παρέχουν λογαριθμικές πιθανοφάνειες/εντροπίες ως *σήματα αβεβαιότητας* και *κριτήρια απόρριψης εκτός κατανομής (OOD)*, με ελάχιστο πρόσθετο κόστος και καλή συμβατότητα με μοντέρνους ανιχνευτές.

1.4 Μεθοδολογία

Η προτεινόμενη προσέγγιση στοχεύει σε *ανοικτή* ανίχνευση αντικειμένων σε σενάρια αέρος–αέρος, με **δύο συμπληρωματικά στάδια**: (i) **Joint Thresholding**, όπου συνδυάζονται η εγγενής εμπιστοσύνη του μοντέλου με την εντροπία των GMM για αποφάσεις διατήρησης/ απόρριψης ανά ανίχνευση, και (ii) **Fusion MLP**, όπου ένα ελαφρύ πολυεπίπεδο perceptron μαθαίνει να συγχωνεύει πολλαπλές ενδείξεις αβεβαιότητας σε έναν ενιαίο, βαθμονομημένο κανόνα απόφασης. Και στα δύο στάδια, ο ανιχνευτής παραμένει αμετάβλητος (*detector-agnostic*) και απαιτείται μόνο πρόσβαση στα εμφωλεύματα (embeddings) και στα logits/σκορ του.

1.4.1 Joint Thresholding

Δομή pipeline. Κάθε είσοδος (εικόνα) επεξεργάζεται από ανιχνευτή πραγματικού χρόνου με συνελικτικό δίκτυο που έχει κανονικοποιηθεί φασματικά (spectral normalization) για ευσταθή εμφωλεύματα. Για κάθε προτεινόμενη ανίχνευση (detection) λαμβάνονται: (α) εγγενή σκορ εμπιστοσύνης (softmax/score confidence, εντροπία, πυκνότητα των logits) και (β) *σήματα από χώρους εμφωλευμάτων* μέσω **Gaussian Mixture Models** (GMMs) που προσαρμόζονται ανά κλάση στα εμφωλεύματα του συνόλου εκπαίδευσης. Από τα GMMs εξάγονται η πυκνότητα και η εντροπία ως σήματα αβεβαιότητας από τις λογαριθμικές πιθανοφάνειες.

Είσοδοι/Έξοδοι. Είσοδοι: ανά ανίχνευση, η εμπιστοσύνη του ανιχνευτή και η εντροπία του GMM. Έξοδος: *δυαδική* απόφαση *keep/reject* ανά κουτί, ώστε να απορρίπτονται OOD/ αμφίβολες ανιχνεύσεις με ελάχιστο κόστος.

GMM & confidence fusion. Η μέθοδος (joint thresholding) εφαρμόζει *διπλό φίλτρο*: ένα κατώφλι στο σκορ του ανιχνευτή και ένα κατώφλι στο GMM-παράγωγο σήμα (εντροπία). Η ανίχνευση διατηρείται όταν *και* τα δύο κριτήρια ικανοποιούνται. Προηγείται *pruning* πολύ χαμηλών σκορ για απομάκρυνση θορυβωδών, πλεονάζουσων ανιχνεύσεων, και *temperature scaling* για ήπια βαθμονόμηση των logits/ πυκνοτήτων. Η διαδικασία είναι απολύτως μετα-επεξεργαστική (χωρίς νέα εκπαίδευση του ανιχνευτή) και διατηρεί ταχύτητα σε πραγματικό χρόνο.

1.4.2 Fusion MLP

Κίνητρο. Οι χειροποίητοι κανόνες κατωφλίωσης δεν αξιοποιούν πλήρως τη συμπληρωματικότητα των σημάτων. Στο δεύτερο στάδιο, μαθαίνουμε *απευθείας* τη συγχώνευση πολλαπλών ενδείξεων αβεβαιότητας ώστε να βελτιωθεί η διάκριση ID/OOD (και, όπου απαιτείται, η ρητή διάκριση {ID, OOD, BG}).

Δομή pipeline. Τρέχουμε τον προεκπαιδευμένο ανιχνευτή σε δεδομένα με *γνωστές* (ID) και *άγνωστες* (OOD) περιπτώσεις και κατασκευάζουμε νέο σύνολο εκπαίδευσης σε *επίπεδο ανίχνευσης*: για κάθε κουτί εξάγουμε διάνυσμα χαρακτηριστικών που μπορεί να περιλαμβάνει score confidence, softmax entropy/density, GMM log-likelihoods/entropies και, προαιρετικά, logits. Κάθε δείγμα επισημαίνεται ως ID, OOD ή BG (background) βάσει αντιστοίχισης με ground truth.

Είσοδοι/Έξοδοι. Είσοδοι: συμπαγή διανύσματα χαρακτηριστικών ανά ανίχνευση (χωρίς ανάγκη πρόσβασης σε εικονοστοιχεία). Έξοδος: λογάριθμοι πιθανοφάνειας (*logits*) από ένα **ελαφρύ MLP** για (α) *δυαδική* απόφαση ID vs. OOD ή (β) *τριταξική* απόφαση {ID, OOD, BG}. Στη συνέχεια εφαρμόζεται κατωφλίωση στα logits για να ικανοποιούνται περιορισμοί *open-set recognition* (π.χ. επιθυμητά false acceptance rates).

Χαρακτηριστικά υλοποίησης. Το MLP είναι μικρού μεγέθους ώστε να εκπαιδεύεται/εκτελείται σε περιορισμένο υλικό, αξιοποιεί την ίδια ροή εξαγωγής χαρακτηριστικών με το πρώτο στάδιο και συνδυάζεται με τις ίδιες ήπιες βαθμονομήσεις (π.χ. temperature scaling). Η επιλογή χαρακτηριστικών γίνεται συντηρητικά (με σκοπό να αποφύγουμε την υπερπροσαρμογή), ενώ η εκπαίδευση μπορεί να προσαρμόζεται στο σενάριο (δυαδικό ή τριταξικό) χωρίς αλλαγές στον ανιχνευτή.

Συνολικά, το **Joint Thresholding** παρέχει άμεση, χαμηλού κόστους ενίσχυση αξιοπιστίας μέσω απλών, αλλά *συζευγμένων* κανόνων, ενώ το **Fusion MLP** μαθαίνει έναν πιο εκφραστικό κανόνα συγχώνευσης που αξιοποιεί πλήρως τα διαθέσιμα σήματα αβεβαιότητας, διατηρώντας τον ρυθμό καρέ και την κλειστού συνόλου ακρίβεια.

1.5 Πειράματα και Αποτελέσματα

1.5.1 Πειραματική Διάταξη

Χρησιμοποιούμε τα AOT (κλειστό σύνολο), AOT-C (συνθετικές αλλοιώσεις), Real Flights (πραγματικές πτήσεις με έντονο domain shift) και COCO-OS (πολύπλοκο, μεγάλης κλίμακας ανοιχτό σύνολο). Η εκπαίδευση γίνεται σε GPU κατηγορίας A10G και οι ελαφρές συνιστώσες (Fusion MLP) είναι συμβατές με CPU. Μετρικές: AUROC, TPR@OSR (5/10/20%), mAP (CS/OS), FPS.

1.5.2 Μέρος I: Joint Thresholding

Ablation σε σήματα αβεβαιότητας

Για να συγκρίνουμε τον *Joint Thresholding* έναντι δημοφιλών συναρτήσεων αυτοπεποίθησης/αβεβαιότητας, χρησιμοποιούμε υποσύνολο του AOT με εικόνες drones ως άγνωστες κλάσεις. Ελέγχουμε score pruning, spectral normalization και temperature scaling σε όλους τους συνδυασμούς. Παρατηρούμε ότι ο συνδυασμός (SN + pruning + scaling) με Joint Thresholding δίνει τη σταθερά καλύτερη διαχωρισιμότητα.

mAP και FPS

Ελέγχουμε την επίδραση της βαθμονόμησης/κατωφλίωσης στην ανίχνευση κλειστού συνόλου και στην ταχύτητα του ανιχνευτή. Η ακρίβεια CS/ OS mAP διατηρείται (ή βελτιώνεται οριακά λόγω αποκοπής χαμηλών σκορ) και ο ρυθμός παραμένει σε πραγματικό χρόνο.

Σύγκριση με Δημοφιλείς Ανιχνευτές

Συγκρίνουμε με entropy thresholding και αντιπροσωπευτικές ανοικτού συνόλου μεθόδους σε AOT-C και Real Flights. Ο αλγόριθμος Joint Thresholding υπερέχει συστηματικά σε AUROC και AUROC_{bd} χωρίς κόστος σε mAP/FPS, δείχνοντας ανθεκτικότητα σε συνθήκες πραγματικής πτήσης.

1.5.3 Μέρος II: Fusion MLP

Ablation διανύσματος εισόδου

Δοκιμάζουμε συνδυασμούς: detector scores, GMM σήματα και βαθμονομημένα logits, αποφεύγοντας υψηλοδιάστατα embeddings. Το συμπαγές διάνυσμα (scores + GMM + logits) αποδίδει καλύτερα και γενικεύει σταθερά.

Δύο κλάσεις (ID/OOD)

Αντιπαραβάλλουμε το *Fusion MLP* με score/entropy/density και το Joint Thresholding σε AOT-C, Real Flights και COCO-OS. Η συγχώνευση υπερισχύει του ανταγωνισμού, κρατώντας mAP και FPS στα επίπεδα του βασικού ανιχνευτή.

Τρεις κλάσεις (ID/OOD/BG)

Αξιολογούμε Fusion MLP έναντι double-thresholding με ρητή διάκριση υποβάθρου. Το MLP μειώνει background ανιχνεύσεις (false positives) και διατηρεί την απόδοση στις ID κλάσεις.

Domain Shift στην εκπαίδευση Fusion

Εξετάζουμε εκπαίδευση MLP με proxy OOD (άλλο drone dataset, COCO, συνθετικά στο feature space) και σταθερό έλεγχο σε Real Flights. Καμία proxy πηγή δεν υποκαθιστά πλήρως in-domain OOD· η χρήση αντιπροσωπευτικών άγνωστων αντικειμένων κρίνεται απαραίτητη για τη διατήρηση των επιδόσεων του αλγορίθμου.

1.6 Συμπεράσματα

Η διπλωματική εργασία αυτή ασχολήθηκε με το πρόβλημα της ανίχνευσης ανοιχτού συνόλου με εκτίμηση αβεβαιότητας στην εναέρια ανίχνευση αντικειμένων, με έμφαση στις προκλήσεις που εμφανίζονται κατά την ανάπτυξη σε UAV. Παρουσιάστηκε ένα μοντέλο-αγνωστικό πλαίσιο που συνδυάζει συμπληρωματικά σήματα αβεβαιότητας, εισάγει ρητή διάκριση ανάμεσα σε *υπόβαθρο* και *άγνωστα αντικείμενα*, και διατηρεί πραγματικό χρόνο λειτουργίας, κατάλληλο για ενσωματωμένα συστήματα. Σε benchmarks και δεδομένα πραγματικών πτήσεων, το πλαίσιο βελτίωσε σταθερά την ανθεκτικότητα ενώ διατήρησε την ακρίβεια του κλειστού συνόλου, αποδεικνύοντας την πρακτική του αξία σε εφαρμογές κρίσιμες για την ασφάλεια.

Συνοπτικά, τα βασικά συμπεράσματα είναι:

- Το **Joint Thresholding** βελτιώνει το AUROC σε σχέση με τις βασικές μεθόδους χωρίς να θυσιάζει mAP ή FPS.
- Το **Fusion MLP** υπερτερεί των χειροποίητων κατωφλίων και προσφέρει καλύτερη τριμερή διάκριση (ID/OOD/Background).
- Η επιλογή κατάλληλων OOD δεδομένων εκπαίδευσης αποδείχθηκε κρίσιμη για τη γενίκευση και τη σταθερότητα των μεθόδων.

Συνολικά, η εργασία συμβάλλει στη βελτίωση της αξιοπιστίας των συστημάτων αντίληψης UAV, μειώνοντας τα υψηλής εμπιστοσύνης σφάλματα και αυξάνοντας την ασφάλεια σε πολύπλοκα αεροπορικά περιβάλλοντα.

Chapter 2

Introduction

Unmanned Aerial Vehicles (UAVs), commonly known as drones, have experienced a significant surge in applications across various sectors, including military operations, real-time monitoring, emergency response, goods delivery, and scientific research. Their versatility and efficiency have made them indispensable tools in both civilian and military contexts. As the deployment of UAVs increases, so does the complexity of the airspace they must navigate. Particularly in urban environments, the risk of mid-air collisions becomes a pressing concern due to the high density of obstacles and other airborne objects. Current regulations require operators to maintain a Visual Line of Sight (VLOS) with their drones, which limits the potential for fully autonomous operations and restricts UAVs from reaching their full capabilities. To address these challenges, Sense and Avoid (SAA) systems have been developed to enable UAVs to detect and autonomously avoid potential collisions. These systems utilize sensors to observe the environment, recognize threats, and make decisions to minimize risks while accomplishing mission objectives. However, existing SAA systems face significant hurdles, particularly in dealing with non-cooperative traffic that does not share positional information and in operating reliably under varying environmental conditions.

Within aerial perception, two related yet distinct problem settings face different practical challenges. **Air-to-air detection** targets other airborne agents (airplanes, helicopters, drones, birds) against largely unstructured backgrounds (sky, clouds, glare). Targets are typically small, distant, and fast-moving, labeled data are scarce, and missed detections are safety-critical. **Air-to-ground detection** focuses on vehicles, vessels, infrastructure, and terrain from a top-down vantage. Here the pain points are extreme scale variation, arbitrary object orientations, clutter and occlusions, and diverse classes under shifting viewpoints, weather, and illumination. Both regimes share the same constraints: real-time inference on SWaP-limited platforms, robustness to adverse conditions and sensor artifacts (blur, noise, compression), and reliable localization of very small objects.

Reliable perception is therefore critical to enabling robust and safe autonomy in UAV operations, especially in complex air-to-air scenarios involving dynamic, non-cooperative targets. Traditional object detection frameworks typically assume closed-set conditions, where the object categories encountered during inference are known *a priori* and adequately represented in the training dataset. However, real-world UAV deployments frequently violate this assumption due to environmental variations, sensor noise, domain shifts, and the inevitable presence of previously unseen or unknown aerial targets. Such violations can significantly degrade detection accuracy and compromise operational safety, underscoring the necessity of robust open-set detection methods capable of reliably identifying and rejecting unknown or ambiguous targets.

Open-set object detection (OSOD) methods aim explicitly at detecting objects belonging to known categories while effectively rejecting unknown instances, ensuring safer autonomous decision-making under uncertainty. Motivated by these limitations, this thesis introduces a robust, uncertainty-aware OSOD framework specifically designed for air-to-air UAV detection scenarios. The framework integrates semantic uncertainty estimation via embedding-space entropy modeling, drawing inspiration from techniques such as Deep Deterministic Uncertainty (DDU) and Gaussian Mixture Modeling-based detection (GMM-Det). To further enhance robustness, spectral normalization is incorporated to stabilize feature geometry and temperature scaling for confidence calibration. At inference, the detector's native softmax confidence is fused with embedding-space uncertainty to keep high-trust detections and discard ambiguous ones, introducing negligible runtime overhead.

The framework is extensively validated using the challenging AOT-C aerial benchmark and real-world flight experiments conducted under diverse operational conditions. Through systematic ablation studies, improvements in open-set discrimination and robustness over strong YOLO-based baselines are demonstrated, while preserving closed-set mAP and real-time throughput (>20 FPS on embedded platforms). Notably, this method achieves substantial performance gains in adverse real-world aerial conditions.

Contributions

- **Model-agnostic, uncertainty-aware OSOD** for air-to-air detection via embedding-space Gaussian mixtures and entropy, fused with detector confidences for per-box keep/reject decisions.
- **Calibration and regularization**: spectral normalization to smooth feature geometry and temperature scaling to improve confidence calibration under shift.
- **Real-time, low-overhead design**: a post-hoc pipeline that preserves embedded-platform throughput.
- **Comprehensive evaluation**: AOT-C and real flight tests demonstrating higher open-set AUROC and stable closed-set accuracy under adverse conditions.

Chapter 3

Background and Related Work

3.1 Aerial Object Detection

3.1.1 Air-to-Air Object Detection

Air-to-air object detection refers to UAVs detecting other airborne objects (e.g. other drones, manned aircraft, birds) using onboard sensors. This capability is critical for *Sense-and-Avoid (SAA)* systems that prevent mid-air collisions in increasingly crowded skies [3]. In civil applications like parcel delivery drones and urban air taxis, reliable aerial object detection enables autonomous collision avoidance to meet safety regulations (which currently require human line-of-sight). It is also vital for UAV swarm coordination (each drone tracking others visually) and for counter-drone systems to detect malicious UAVs entering protected airspace [3]. These use cases demand real-time performance and high detection reliability, as missed detections or false alarms can lead to accidents in safety-critical scenarios.

Early work on air-to-air detection was limited by scarce specialized datasets. Recently, dedicated benchmarks have emerged to advance this field. Notably, the Airborne Object Tracking (AOT) dataset and similar UAV-to-UAV detection datasets (e.g. the UAV-DetFly and UAV-Fly datasets) have been introduced, providing labeled imagery of flying objects captured from drones [3]. These benchmarks enable standardized evaluation of algorithms for detecting and tracking airborne targets. However, they remain constrained in environmental diversity – e.g. most images are collected in clear weather and uncluttered backgrounds. Thus, current detectors trained on such data may struggle under adverse conditions not represented in the training set.

Key challenges in air-to-air detection include the typically small object size and distant range of targets, as well as relative speeds that can be very high (both the observing UAV and target may be moving rapidly). Airborne objects often appear as barely a few pixels or a tiny silhouette against the sky, making them difficult to distinguish. The background can vary from a plain sky to a complex ground backdrop if the camera's line-of-sight extends toward the horizon. The lack of cooperation from targets (non-cooperative traffic) means no prior information (like GPS broadcast) is available, so detection must rely purely on sensor data. This visual detection approach is one of the few viable options for air-to-air scenarios due to payload limits on small UAVs (which preclude heavy sensors). Overall, air-to-air object detection must achieve high sensitivity to small, fast-moving objects and do so with minimal latency to be useful in collision avoidance.

3.1.2 Air-to-Ground Object Detection

Air-to-ground object detection covers scenarios where aerial platforms (UAVs, aircraft, or satellites) detect objects on the ground. This is common in remote sensing and surveillance applications, such as traffic monitoring, search-and-rescue, precision agriculture, and military reconnaissance. Unlike air-to-air, here the camera looks downward to identify targets like vehicles, ships, buildings, or people on the Earth's surface. Large-scale datasets like **DOTA** [43] (Dataset for Object Detection in Aerial images) and **FAIR1M** [39] (Fine-grained Object Recognition in Remote Sensing Imagery) have driven progress in this area. For instance, the DOTA dataset contains over 2,800 high-resolution aerial images (around 4000×4000 pixels each) with about 188,000 annotated object instances across 15 categories. Each object is labeled with an oriented bounding box (a quadrilateral), reflecting the fact that in aerial views objects can appear at arbitrary orientations rather than aligned to the image axes. FAIR1M is an even larger benchmark with over 15,000 images and 1 million object instances, focusing on fine-grained categories of aircraft, ships, and vehicles. These datasets highlight unique challenges of air-to-ground vision: enormous scale variations (a single image can contain both close-up large objects and far-away tiny objects), densely crowded scenes in some areas and sparse regions in others, and the need to handle rotated objects and different viewpoints.

Cutting-edge models for aerial image detection have adapted general object detection frameworks to meet these challenges. Many two-stage detectors (e.g. Faster R-CNN) and one-stage detectors (e.g. YOLO) have been evaluated on DOTA and FAIR1M, often with modifications for rotated bounding boxes (such as oriented R-CNN or transformer-based methods for rotation). Results show that while deep learning detectors achieve good accuracy on these benchmarks, performance can still lag behind that on natural image datasets like COCO, due to the increased complexity of aerial scenes. Nonetheless, continuous improvements are being made: for example, specialized oriented-object detectors (like the recent AO2-DETR transformer [8]) have attained state-of-the-art results by directly predicting rotated boxes and accounting for orientation in their design. Air-to-ground detection remains an active research area, bridging computer vision and remote sensing, with importance for both civilian and defense-related applications.

3.1.3 Common Challenges in Aerial Detection

Despite the differing perspectives, air-to-air and air-to-ground detection share many common constraints that make the problem particularly challenging:

- **Small Object Size and Scale Variation**: Aerial objects often occupy only a few pixels or a tiny fraction of the image (e.g., a small drone at distance, or a vehicle in a wide-area satellite image). This makes detection difficult, as models must discern minute targets from background noise. Extreme scale variation can occur, requiring detectors to handle both very large and very small instances in the same frame.

- High Speeds and Dynamic Scenes: In air-to-air scenarios, both the sensor platform and the target may be moving at high velocity, drastically reducing the time window for detection and increasing motion blur. Even in air-to-ground settings, a UAV moving at speed introduces motion parallax and rapidly changing viewpoints. Fast-moving objects (e.g. another UAV or a car) exacerbate the challenge of obtaining a clear detection in time. Dynamic backgrounds (moving clouds, swaying trees, etc.) can further complicate distinguishing true objects.

- **Occlusion and Clutter**: Aerial images can be cluttered with many irrelevant objects or textures (waves on water, patterns on the ground). In urban environments, ground objects can be partially occluded by buildings or vegetation. Similarly, multiple airborne objects might overlap in the camera view. Such occlusions and background clutter make it hard for detectors to isolate the object of interest from the background.

- **Varying Environmental Conditions**: Aerial detection must contend with changing lighting (day/night, sun angles), weather conditions (fog, rain, snow), and atmospheric effects. For example, fog or haze can obscure distant objects, and glare can wash out camera images. These factors lead to domain shifts between training data (often collected in clear, favorable conditions) and real-world deployment scenarios. Without robustness to such variations, models that perform well in the lab can fail in the field.

- **Sensor and Platform Variability**: Different UAVs may use different cameras or sensors (with varying resolutions, fields of view, spectral bands, etc.), and capture data from different altitudes or angles. A model trained on one sensor's data may not directly generalize to another sensor due to differences in image characteristics (color, noise profile, etc.). Moreover, as UAVs move, the viewpoint can change rapidly (e.g., looking forward versus downward), causing the appearance of objects to shift.

- **Real-Time Processing Requirements**: Both air-to-air and air-to-ground detection often require real-time operation. In collision avoidance, decisions must be made within seconds or less to be effective. This imposes strict latency and computational constraints on detection algorithms. The UAV's onboard computer typically has limited processing power due to Size, Weight, and Power (SWaP) constraints, so the detection method must be efficient. High accuracy is needed, but not at the expense of speed – the system must maintain a high frame rate to track fast events. Achieving robustness under the above challenges while meeting real-time and hardware limitations is a core difficulty in aerial object detection.

3.2 The Collision Avoidance Pipeline

A UAV's collision avoidance system can be conceptualized as a multi-stage pipeline that mimics a human pilot's process of perceiving and reacting to threats. This Sense-and-Avoid pipeline takes raw sensor inputs and produces safe navigational actions. We provide an analytical overview of the pipeline, outlining each stage and the associated methods, as depicted in recent literature. We also discuss the sensing modalities available, the detection and tracking algorithms used to perceive obstacles, the decision-making approaches to select avoidance maneuvers, and the various system constraints that influence the pipeline design.

3.2.1 Pipeline Stages Overview

The collision avoidance pipeline typically consists of four key stages:

1. **Sensor Data Acquisition (and Preprocessing)**: The system first gathers observations of the environment through onboard sensors (camera frames, LiDAR point clouds, radar signals, etc.). Data augmentation or preprocessing may also occur at this stage to enhance sensor inputs. This raw sensor feed provides the basis for all subsequent analysis.

2. **Object Detection**: Next, algorithms analyze the sensor data to detect and localize objects that could pose collision threats. In this stage, the system differentiates foreground objects from the background and estimates their bounding boxes or other descriptors. Importantly, this detection step is performed on a frame-by-frame basis (a "temporal snapshot"), not yet linking observations over time. The output is a set of observed objects (obstacles or other aircraft) with their positions (and sometimes categories or sizes).

3. **Re-Identification and Tracking**: The detected objects are then fed into a tracking module that associates detections across consecutive time frames. Each newly detected object is either matched to a previously seen object (re-identification) or initialized as a new track if it hasn't been seen before. Through tracking, the system maintains a situational picture of each threat's trajectory (position over time and estimated velocity). This spatio-temporal modeling of threats greatly aids prediction and decision-making.

4. **Decision-Making Algorithms**: Finally, given the detected and tracked obstacles and their predicted trajectories, the system must decide on an avoidance maneuver (or confirm that none is needed). The decision module takes the world model from previous stages and computes the optimal evasive action to avoid collision, balancing safety as the top priority with other objectives like mission continuity or energy efficiency. This could involve choosing a new flight path, adjusting speed, or in multi-rotor drones, an immediate evasive maneuver.

These stages operate in a closed loop at high frequency, constantly sensing, detecting, tracking, and updating decisions as the UAV moves. In practice, there may be feedback between stages (e.g., the decision to maneuver might reset tracking of an object if the ownship UAV turns abruptly). Each stage has to be robust and efficient for the overall system to function reliably in real-time.

3.2.2 Sensing Modalities: Cooperative vs. Non-Cooperative

The foundation of the SAA pipeline is the sensor suite. Broadly, sensing approaches for collision avoidance are categorized into cooperative and non-cooperative modalities:

- Cooperative Sensing: These methods rely on cooperative communication between aircraft. For example, transponders and systems like ADS-B (Automatic Dependent Surveillance–Broadcast) allow aircraft to broadcast their own GPS position, velocity, and ID to others. If all aircraft in an airspace are equipped and enabled, a UAV can simply receive positional data of nearby traffic and predict collisions. Cooperative sensors (including Traffic Collision Avoidance Systems, TCAS) are effective in scenarios where adoption is wide. However, they fail to detect "non-cooperative" objects that do not broadcast signals (e.g., a hobby drone or bird). They also require extra communication hardware and have reliance on spectrum and protocol compliance.

- Non-Cooperative Sensing: To handle objects that do not self-report their position, UAVs use onboard sensors to perceive the environment. A variety of sensor types have been explored, each with pros and cons:

  - Vision Cameras: Normal RGB cameras are popular due to their lightweight, low cost, and richness of information. They can detect a wide range of object types and provide classification cues. The downside is that interpreting images is computationally intensive – requiring advanced computer vision algorithms – and performance can degrade in poor lighting or weather.

  - LiDAR: Laser scanners provide precise 3D distance measurements to obstacles. LiDAR can directly sense range and shape, which is advantageous for accurate obstacle localization. However, LiDAR units are typically heavier and more power-hungry, and their performance is severely impacted by weather (rain, fog, and dust can scatter the laser). Cost is also a limiting factor for many UAV applications.

  - Thermal Infrared Cameras: These can detect heat signatures and so might pick up other aircraft via engine heat or warm bodies (for birds). Yet, thermal sensors have limited range and resolution, and like vision, can be affected by weather obscurants. They are more often used in low-light/night scenarios as a complement to visible cameras.

  - Radio-Frequency (RF) Sensors: RF-based ranging (like radar altimeters or passive RF detectors) can detect objects emitting radio signals or radar reflections. They can work at long range, but RF methods are prone to interference and noise in cluttered electromagnetic environments. For example, a drone might use a simple radar to detect large obstacles, but small drones or birds provide very weak radar returns and may be missed.

  - Millimeter-Wave Radar: Specialized automotive-style radars operating at high frequency (e.g., 77GHz) have been tested on drones. They are all-weather and can detect objects' range and relative velocity. However, distinguishing small airborne objects with radar is challenging, as is resolving closely spaced objects. Moreover, radars add to system complexity and require significant processing.

Each UAV collision avoidance system must carefully select a sensor suite that balances these trade-offs. Often, a combination is used (sensor fusion), e.g. a camera for object classification and a radar for range measurement. Ultimately, the sensing modality sets the

stage for the detection algorithms and influences their design (e.g., vision-based detectors vs. LiDAR-based obstacle detection).

3.2.3 Object Detection and Tracking in SAA

Once sensor data is acquired, the pipeline's perception components (detection and tracking) identify and characterize airborne objects. Object detection in SAA systems is often implemented with advanced computer vision models that take images (or other sensor data) and output bounding boxes around potential obstacles. Recent research has applied deep learning detectors, originally developed for generic object detection, to the aerial domain. For instance, convolutional neural network (CNN) based detectors have been trained to spot other UAVs in camera images. A notable example is the work by Arsenos *et al.* [**?**], who developed a vision-only real-time collision avoidance framework using object detection, tracking, and distance estimation with deep learning models. In their pipeline, a YOLO-based detector was used to identify flying objects in each frame, and then stereo vision techniques estimated range – demonstrating a purely vision-driven SAA solution.

Detection in this context must prioritize high recall (not missing any true obstacle) while maintaining low false alarms. Some approaches tailor the detection algorithms to aerial scenarios, for example by training on drone images and augmenting data to simulate various backgrounds (sky, clouds, ground). The output of the detector (locations of objects) is then passed to the tracking stage. Tracking algorithms (like Kalman filters, SORT, or deep SORT for vision-based tracking) take these raw detections and link them over time to form trajectories. By tracking, the system can estimate the relative velocity and heading of a detected object, which are crucial for predicting future positions and collision risk assessment. Tracking also helps smooth out noise from the detector (since a consistent track gives more confidence than a single-frame detection) and can handle brief occlusions by predicting where an object will reappear.

A challenge in aerial tracking is maintaining locks on very small objects that may maneuver quickly. Approaches to improve robustness include using the physics of flight (e.g., assuming a flying object will follow certain motion constraints) or employing re-identification descriptors (appearance features of the object) to avoid identity switches. The end product of the detection & tracking stage is a situational awareness: the UAV knows "what" is around it (object detections) and "where it is going" (tracks and velocities of those objects).

3.2.4 Decision-Making Algorithms for Avoidance

With a dynamic world model from the tracker, the final pipeline stage is to decide on avoidance maneuvers. Over the years, various decision-making paradigms have been explored, ranging from simple rule-based strategies to sophisticated machine learning methods:

- **Rule-Based and Geometric Methods**: Traditional collision avoidance logic often uses predefined rules or geometric calculations. An example is the Probabilistic Intersection Collision Avoidance (PICA) algorithm, a rule-based method that uses current distances and relative velocities to decide evasion, without needing any learning. Such algorithms tend to be computationally lightweight and can be analytically verified for safety. However, they might be myopic (considering only current snapshot, not future trajectory) and have limited adaptability to complex scenarios.

Legacy systems like the ACAS XO advisory for manned aircraft rely on massive lookup tables of optimal maneuvers for given states, but these tables can be too large for UAVs (several GB in size). Researchers have compressed these into neural networks – e.g., using a deep network to approximate the table with much smaller memory (tens of MB) – merging rule-based optimal strategies with learning-based function approximation.

- **Reinforcement Learning (RL) Approaches**: Deep reinforcement learning allows a UAV to learn collision avoidance policies by trial-and-error in simulation. For instance, Ouahouah *et al.* propose a DQN-based approach named RELIANCE, which learns an avoidance policy by interacting with simulated UAV traffic. The RL agent observes the state (positions of threats) and outputs maneuver actions, trained to minimize collision incidents. RL methods can in theory handle complex, dynamic environments by learning adaptive strategies. In practice, they require extensive training and careful reward design, and the learned policy's reliability outside its training conditions can be hard to guarantee. RELIANCE was shown to outperform the rule-based PICA in dynamic scenarios (since it anticipates future behavior by learning from data), but it demands more computational resources (e.g., needing a GPU or powerful onboard computer for real-time inference).

- **Imitation Learning**: Another avenue is training the UAV's decision module by imitating an expert (e.g., human pilot or a known optimal strategy). The system learns a mapping from sensor/tracker inputs to avoidance actions by observing the decisions of an expert in many scenarios. This can achieve good performance if the expert demonstrations cover diverse cases. One challenge found in practice is that the camera's field of view can be narrow, so an imitation policy might not generalize if an obstacle approaches from an angle outside the training scenarios. Also, a broader field of view (or multiple cameras) could improve safety but at the cost of more data processing and possibly lower image resolution per camera. Thus, imitation-learned policies must carefully balance sensor configuration with learned behavior.

In summary, the decision-making stage can be implemented via a spectrum of methods: simple reactive rules for resource-limited drones, or complex learned policies (RL or imitation) for drones with more computing capability and in more unpredictable environments. Hybrid approaches also exist, such as using rule-based logic for well-understood cases and a learned policy for edge cases, or running a learned policy with a rule-based safety override. Regardless of approach, any decision algorithm for SAA must be thoroughly tested to ensure it avoids collisions reliably and does not introduce unsafe maneuvers (the evaluation often happens first in high-fidelity simulations and then in controlled flight tests).

3.2.5 System Constraints and Considerations

Real-world UAV collision avoidance must operate under stringent system constraints:

- **Onboard Hardware (SWaP) Limitations**: Small UAVs are constrained in Size, Weight, and Power. This limits the processing hardware that can be carried – often to a lightweight embedded GPU or CPU. As a result, algorithms that are too computationally heavy or memory-intensive may be impractical for real-time use. The pipeline must be optimized to run within the available computing budget, sometimes

requiring quantized models, efficient neural network architectures, or offloading computation to edge servers when possible. However, offloading data comes with its own issues, as described next.

- **Communication Latency and Reliability**: If parts of the processing (especially the detection/tracking) are offloaded to a ground station or cloud (to leverage more powerful computing), the communication link's latency and stability become critical. Network delays or dropouts could render the avoidance system ineffective when a fast reaction is needed. For this reason, many SAA systems strive to be self-contained onboard. When wireless links are used (for example, to get cooperative traffic info or to offload heavy vision processing), designers must account for potential lag and packet loss. Techniques like prediction buffers can mitigate latency (by planning avoidance assuming worst-case delays), but the safest course is often to require autonomy without reliance on real-time comm links.

- **Real-Time Operation**: Collision avoidance is inherently a real-time task – detection - to -decision loops must execute within fractions of a second to be useful for high-speed UAVs. This imposes a hard constraint on each pipeline stage's execution time. For instance, if a UAV is closing in at 20 m/s to an obstacle and needs at least 1 second to safely turn, the system must detect and initiate an avoidance maneuver at least a second in advance. That leaves very little margin for computing. Every stage from sensor readout, inference (detection/tracking), to decision-making must be streamlined and possibly run in parallel or on dedicated hardware accelerators.

- **Safety and Redundancy**: As a safety-critical system, the collision avoidance pipeline often includes redundancies and fail-safes. Multiple sensors can provide redundancy (e.g., having both a camera and a radar – if one fails or is uncertain, the other can confirm). The decision logic might incorporate safety margins – for instance, issuing avoidance commands that err on the side of caution if any uncertainty exists in the object state estimation. Furthermore, rigorous validation (often following aviation standards) is needed. These considerations can sometimes conflict with performance; for example, adding more sensors improves reliability but adds weight and processing load.

Given these constraints, researchers have emphasized the need for algorithms that are not only accurate but also resource-efficient. Recent studies highlight the incompatibility of certain advanced techniques with UAV platforms: for instance, some domain generalization or deep learning methods dramatically improve robustness but are too slow or heavy for onboard use. Thus, an active area of research is slim, optimized neural networks and algorithms that maintain high detection and avoidance performance while fitting within the tight SWaP and real-time envelope of UAV systems.

3.3 Vision Models for Aerial Detection

Having reviewed the pipeline and sensing, we now focus on the core vision models used for aerial object detection. Modern object detectors can be categorized by their architecture into one-stage and two-stage approaches, each with implications for performance and suitability in aerial tasks. We highlight representative models from both categories – particularly the evolution of the YOLO family, and transformer-based detectors like DETR – and note their performance on standard benchmarks (COCO for generic objects, and DOTA/FAIR1M for aerial images).

3.3.1 One-Stage vs. Two-Stage Detectors

In traditional computer vision, two-stage detectors (e.g., the R-CNN family) first generate region proposals and then classify each proposal, whereas one-stage detectors (e.g., YOLO, SSD) predict class probabilities and bounding boxes in a single pass over the image. Two-stage models historically achieved higher accuracy on benchmarks by focusing on a few promising object regions, but at the cost of speed. One-stage models are designed for speed, performing dense prediction over the image in one go, but had to close the accuracy gap with innovations in network design and loss functions.

In the aerial domain, the trade-offs manifest in specific ways. Because aerial images often contain many small objects and the objects can appear in arbitrary orientations, the detector's design needs to handle those. Two-stage detectors can be adapted (e.g., Faster R-CNN with rotated RoI pooling to handle oriented boxes). One-stage detectors can incorporate custom anchor boxes or prediction heads for small and rotated objects. An important consideration is that many aerial platforms (like drones) demand real-time inference, favoring one-stage methods for their efficiency. Indeed, recent evaluations have found that advanced one-stage models can match or exceed two-stage models in accuracy while being faster, making them attractive for UAV use. For example, Arsenos *et al.* report that one-stage YOLO models not only run in real-time but also exhibit better robustness under image corruptions compared to a two-stage Faster R-CNN in air-to-air detection tasks. In contrast, transformer-based detectors (which we discuss below) and older two-stage methods showed vulnerability to domain shifts, despite competitive standard accuracy.

3.3.2 Representative Detection Models

YOLO Series (One-Stage): The "You Only Look Once" (YOLO) family of models has become synonymous with real-time object detection. From YOLOv1 (2016) through YOLOv8 (2023) and beyond, each iteration has improved on accuracy and capabilities while preserving high speed. YOLO models predict bounding boxes and class probabilities through a single forward network pass, making them extremely fast – a crucial advantage for onboard UAV deployment. They also tend to be lightweight, which aligns with UAV hardware limits. In aerial tasks, YOLO has been widely used; for instance, as a baseline in open-set drone detection research, YOLOv5 achieved around 40–43 mAP on known object classes in a UAV image benchmark. However, a standard YOLO trained on a fixed set of classes will treat any object as one of those classes – it has no built-in mechanism to recognize an object it was never trained to detect (the open-set problem). This can lead to false detections when an unknown object appears. Recent work addresses this by adding uncertainty estimation to YOLO-like models, so that the detector can say "I'm not sure

what this is" for novel objects. The most up-to-date YOLO versions continue to push the envelope, incorporating transformer layers and other enhancements. On the COCO dataset, top YOLO models reach around 50% AP (average precision) while running at 60+ FPS on GPU – a balance of accuracy and speed that is hard to beat. On aerial datasets like DOTA, YOLO models adapted for oriented boxes have also been successful, though they may require adjustments (such as angle prediction) to handle the rotated targets common in aerial images.

DETR and Transformer-Based Detectors: A major recent development in object detection is the introduction of Transformers, as exemplified by DETR (Detection Transformer). DETR reframes detection as a direct set prediction problem using an encoder-decoder transformer architecture, eliminating the need for hand-designed anchor boxes and non-maximum suppression. The original DETR achieved comparable accuracy to Faster R-CNN on COCO (around 42% AP) but needed long training schedules and struggled with small objects. Subsequent variants (Deformable DETR, SMCA, etc.) improved training speed and small object detection. In aerial imagery, the oriented-object extension of DETR has shown promise. For example, Rotated DETR and AO2-DETR introduce mechanisms to predict rotated boxes by generating oriented proposals and rotation-invariant features. These models directly output angled bounding boxes suitable for datasets like DOTA and FAIR1M. The benefit of a transformer approach is that it can globally reason about the image context, potentially handling dense scenes or learning long-range dependencies (like groups of objects). However, transformers tend to be heavy; the computational load of DETR can be significant, and early studies (like Michaelis *et al.* for autonomous driving) found that transformer detectors could be less robust to distribution shifts. For UAV applications, real-time performance is a concern – a naive DETR model might not meet the frame rate requirement on onboard hardware. Research is ongoing to compress and accelerate such models, or hybridize them with convolutional features to get the best of both worlds.

Oriented Object Detectors for Remote Sensing: Because standard COCO-style detectors output axis-aligned boxes, a special class of detectors has been devised for aerial image tasks requiring oriented bounding boxes. These include adaptations of classic models (e.g., Oriented Fast R-CNN, Rotated RetinaNet) and bespoke architectures. Many oriented detectors introduce additional angle predictors and rotation-aware loss functions. For instance, one approach is to predict the four vertices of the bounding box polygon instead of just width/height/angle, thus capturing orientation implicitly. DOTA benchmark results over the years show steady improvement: early methods achieved around 60% mAP, while more recent ones with deeper backbones and ensemble strategies exceed 80% mAP on DOTA-v1.0. DETR-OBB (oriented bounding box) variants combine transformers with angle prediction to reach state-of-the-art. In the FAIR1M dataset (which has numerous fine-grained classes like different aircraft models or ship types), detectors often incorporate a classification head that can handle many categories and subtle differences – sometimes using a two-step process (detect the object, then a secondary classifier for fine-grained recognition). A noteworthy observation is that detectors which perform best on natural images are not always the top on aerial images; specialized models or training techniques are needed to cope with the tiny objects and extreme aspect ratios (for example, a runway aligned airplane can be a very slender, rotated bounding box). Nevertheless, progress in generic object detection has greatly benefited aerial detection: techniques like data augmentation, multi-scale feature pyramids, and better backbone networks (e.g., ResNeXt, Swin Transformer backbones) have all been transferred to the aerial domain to boost performance.

In summary, the state-of-the-art vision models for aerial object detection include fast one-stage models (YOLO and its variants) that excel in real-time settings, and advanced two-stage or transformer-based models (including DETR and oriented detectors) that achieve high accuracy and are tailored to aerial images. On benchmarks, one-stage models offer a compelling accuracy-speed tradeoff (for example, YOLO-based models are often top performers in real-time UAV tracking competitions), while transformer models promise improved long-range context understanding and end-to-end simplicity (no post-processing) at the cost of increased computation.

3.4 Domain Generalization

As UAVs venture into varied environments, a trained model will inevitably encounter conditions different from its training data, causing a drop in performance. This problem is referred to as domain shift: the statistical distribution of input data (and possibly labels) changes between training (source domain) and deployment (target domain). For example, a drone vision model trained on sunny daytime images may falter on cloudy or twilight conditions; a detector trained on simulation or one geographic region may not work well in a new locale. Traditional approaches to address this involve transfer learning or domain adaptation, where some data from the target domain is used to fine-tune the model. However, in many UAV scenarios, collecting or labeling target-domain data (e.g., every possible weather or location) is impractical. This motivates the concept of Domain Generalization (DG), where the aim is to train models that generalize to unseen domains without any target domain exposure during training.

In formal terms, domain generalization techniques assume multiple source domains available during training (e.g., images captured in different cities, seasons, or sensor settings) and seek to learn a model whose performance will remain high on a new domain drawn from the same task. Unlike domain adaptation, no target domain data (not even unlabeled) is used in training – the model must be inherently robust to shifts. DG also differs from standard transfer learning: instead of simply fine-tuning on a new domain, DG prepares the model to handle new domains from the outset. Essentially, DG attempts to capture the invariances and essential features of the task that hold across domains, and to avoid overfitting to domain-specific cues.

Research in domain generalization has proposed a variety of methods. These can be grouped into three broad categories:

1. Data Manipulation Techniques: Methods that enrich or manipulate the training data to expose the model to a wider variety of conditions. This includes data augmentation (applying transformations like random crops, flips, color changes, adding noise) beyond the typical, sometimes in a learned or adversarial manner to simulate domain shifts. For example, one could randomize image styles or weather conditions (known as domain randomization) so the model learns to rely on invariant features. Another approach is data generation, where additional training samples are synthesized (e.g., using GANs or neural style transfer) to cover hypothetical domains. The goal is to have training data that is as diverse as possible, so that any new domain appears as just another variation the model has seen. In UAV context, this might mean augmenting images to simulate different sensor noise levels, motion blur, lighting, or backgrounds (forest vs urban scenes) to prevent the model from latching onto spurious domain-specific details.

2. Representation Learning Techniques: These methods focus on learning features that are domain-invariant – i.e., the model's internal representations generalize across domains. One popular strategy is domain adversarial training (as in DANN frameworks), where the model is trained to perform the task (e.g. detection) while simultaneously trying to confuse a domain discriminator network that attempts to predict which domain a sample comes from. By adversarially learning, the feature extractor is encouraged to remove domain-specific information. Other representation approaches include invariant risk minimization (finding features that have a stable correlation with labels across domains) and feature disentanglement (separating features into domain-specific and domain-shared parts and only using the latter

for prediction). Contrastive learning can also be used: e.g., ensure that an object's features from domain A are close to the same object's features from domain B, to enforce domain invariance. In aerial vision, an example might be training a detector on both simulated and real images such that the internal feature maps for an airplane are similar regardless of sim vs real, thus bridging the gap without explicit adaptation.

3. Learning Strategies and Meta-Learning: These methods alter the training procedure or learning strategy to promote generalization. Ensemble learning is one, where multiple models (or multiple specialized classifiers) are trained on different domains or subsets and then combined, so that a new domain can be handled by some mixture of these experts. Meta-learning (learning-to-learn) approaches explicitly simulate domain shift during training: the training data is split into pseudo-train and pseudo-test with different "domains," and the model is optimized to do well on a new pseudo-test domain after seeing the pseudo-train domains. This forces the model to acquire a generalization ability. Approaches like MLDG and MetaReg train models in an episodic fashion to be ready for domain changes. Other strategies include gradient-based methods (adjusting or regularizing the optimization process so that it doesn't overfit source domains) and self-supervised auxiliary tasks (which improve the learned features' generality). In essence, these approaches treat domain generalization as a problem of learning robust training routines that yield models capable of extrapolation.

It is worth noting that these categories are not mutually exclusive – a practical DG approach for UAV detection might combine data augmentation with an invariant feature loss, for example. The surveys by Zhou *et al.* and Wang *et al.* provide comprehensive overviews of these techniques, indicating that combining complementary DG methods often yields the best results.

While most domain generalization research is demonstrated on tasks like image classification, its relevance to aerial object detection is growing. UAVs frequently face train-test domain gaps: e.g., a model trained on one geographic region's imagery might be deployed globally, or a model trained in simulation deployed in reality. In such cases, DG methods can improve reliability. For instance, Arsenos *et al.* introduced common corruption benchmarks for air-to-air detection and showed that training with simulated corruptions (a form of data augmentation for DG) improved detectors' real-world robustness. Generally, applying DG to aerial vision means accounting for things like different camera lenses, different altitudes or viewpoints, seasonal changes in landscapes, or unpredictable lighting/weather – all without having examples of every case in training. Our later chapters will delve deeper into how domain generalization techniques can be tailored and applied to UAV open-set object detection. For now, we acknowledge that DG offers a pathway to enhance model robustness against domain shifts that are inherent in any practical UAV deployment, complementing other approaches like domain adaptation when target data is available.

3.5 Summary and Research Gap

In this chapter, we surveyed the landscape of aerial object detection and the supporting technologies for UAV collision avoidance, and we reviewed the concept of domain generalization as a solution to robustness challenges. We highlighted that air-to-air object detection is an essential component for UAV sense-and-avoid systems, with stringent real-time and reliability requirements given the safety-critical nature of mid-air collision avoidance. We also described air-to-ground detection in remote sensing, noting analogous challenges in detecting small, oriented objects from aerial views. Common obstacles such as tiny object size, high relative speeds, occlusions, and environmental variability make both tasks extremely challenging. The SAA pipeline was broken down into sensing, detection, tracking, and decision stages – each of which has seen extensive research. Modern UAVs can leverage a range of sensors (cooperative ADS-B, onboard vision, LiDAR, etc.), and advanced detection/tracking algorithms (like deep learning-based vision detectors) to build situational awareness. For decision-making, both classical rule-based methods and modern learning-based methods (DQN, imitation learning) are being investigated, with trade-offs in adaptability vs. computational cost. Throughout, we emphasized the system constraints (limited compute, need for real-time, communication limits) that force practical solutions to be efficient and robust.

We then reviewed state-of-the-art vision models for detection, noting that one-stage detectors (exemplified by the YOLO family) and two-stage/transformer detectors (exemplified by Faster R-CNN and DETR, including oriented detectors) each have roles to play. One-stage detectors shine in real-time performance and have been successfully deployed on drones, though they historically assume a closed-set of object classes. Transformer-based detectors offer a new paradigm with potential accuracy and robustness gains, but can be heavy for UAV use. Empirical evaluations in the literature suggest that, under ideal conditions, many detectors can achieve high accuracy, but under shifting or corrupted conditions relevant to UAV flight, their performance can degrade significantly. This naturally led to the discussion of domain generalization, where we saw that a plethora of methods exist to tackle unseen domain shifts by augmenting data, learning invariant features, or adopting special training regimes. Domain generalization is particularly pertinent to UAV applications, because a UAV may encounter novel environments (unseen backgrounds, weather, or sensor settings) without warning.

After assessing the body of prior work, we identify a pressing research gap at the intersection of these topics: the lack of a robust, real-time open-set object detection (OSOD) system integrated into UAV collision avoidance pipelines under domain shift conditions. In other words, current UAV vision systems do not adequately handle objects outside of their trained categories (the open-set problem) especially when the operational domain differs from the training domain. Standard object detectors will confidently classify or ignore an unknown object, which is dangerous in an SAA context (e.g., mistaking a new type of drone for a known bird, or missing it altogether). Likewise, domain shifts (like sudden fog, or a different cityscape) can cause detectors to fail to detect even known objects. While research in open-set detection and domain robustness exists in general computer vision, very few works have applied it to the aerial domain and none, to our knowledge, provide a complete real-time solution suitable for onboard UAV deployment. The recent study by Loukovitis *et al.* took steps in this direction by adding uncertainty-based OOD scoring to a real-time detector, improving its ability to recognize when it sees an unfamiliar object. However, this is one of the first of its kind, and it underscores how nascent this area is. Even their approach, while promising, highlights the complexity of balancing multiple

uncertainty measures and maintaining detection accuracy and speed.

In conclusion, the background and related work point to the need for a new approach that marries high-performance object detection with open-set recognition and domain generalization, tailored for UAV constraints. The remainder of this thesis will address this gap. We aim to design and evaluate a framework for open-set aerial object detection that remains reliable under a range of domain shifts and meets real-time operational requirements. By doing so, we hope to advance UAV sense-and-avoid capabilities toward safer autonomous flight in unstructured, real-world environments, where the only constant is uncertainty.

Chapter 4

Theoretical Background

4.1 Uncertainty in Machine Learning: Aleatoric vs. Epistemic

Uncertainty quantification is increasingly recognized as a critical component in modern machine learning, especially for safety-critical systems such as UAVs. Broadly, uncertainty refers to how confident a model is in its predictions, and more precisely, **what kinds of unknowns** are contributing to that lack of confidence. In the literature, a common decomposition is into two principal types of uncertainty: **aleatoric** and **epistemic**.

4.1.1 Definitions and Conceptual Distinction

Aleatoric uncertainty (also called data uncertainty) refers to uncertainty inherent in the observations. It stems from noisy, ambiguous, or incomplete data. Examples include measurement noise, ambiguous labels, inherent overlap between classes, or sensor/environmental effects such as fog, low light, or occlusion. Importantly, aleatoric uncertainty is often **irreducible** in the sense that collecting more data does not always eliminate it, since the underlying phenomenon may be inherently stochastic.

Epistemic uncertainty (also called model uncertainty) arises from a lack of knowledge about the appropriate model. This could be due to insufficient training data, model misspecification, uncertainty over model parameters, or overconfidence in regions where the model has not seen sufficient examples. Epistemic uncertainty **can** be reduced by more or better data, by more expressive or better-trained models, or by architectural or algorithmic improvements.

As discussed by Hüllermeier and Waegeman [18], supervised learning predictions can be understood as being affected by both sources: uncertainty in the data generation process (aleatoric) and uncertainty in the learned model parameters or hypothesis (epistemic).

4.1.2 Measurement and Estimation Techniques

Several common methods have been proposed to estimate these uncertainties:

- **Ensembles**: training multiple models with different initializations or subsets of data and measuring the disagreement among outputs. Disagreement captures epistemic uncertainty, while averaging provides predictive uncertainty. Ensembles are often found to be among the most reliable methods [40].

- **Monte Carlo Dropout and Bayesian methods**: introducing stochasticity into model parameters at inference time (e.g., through dropout) approximates uncertainty over parameters and is useful for estimating epistemic uncertainty [9]. Extensions also allow capturing aleatoric uncertainty by predicting variance alongside outputs.

- **Loss functions with variance prediction**: in regression tasks, a network may predict both mean and variance, with the loss designed so that the variance models aleatoric uncertainty. In classification, aleatoric uncertainty can be approximated via confidence scores such as softmax probabilities [21].

- **Evaluation strategies**: distinguishing aleatoric and epistemic uncertainty often relies on controlled tests, for example by comparing the stability of aleatoric estimates versus the variability of epistemic ones across regions of the input space [40].

4.1.3 Applications in Object Detection

While much of the foundational research has focused on classification and regression, uncertainty estimation has also been applied to object detection and segmentation, where the problem is more complex due to multiple simultaneous outputs (localization and classification).

For example, Liu et al. [26] study aleatoric uncertainty in camouflaged object detection, where ambiguous signals from low contrast or environmental noise contribute strongly to uncertainty. In such settings, explicitly modeling aleatoric uncertainty helps reduce false positives and mitigate overconfidence.

These applications highlight an important challenge: object detection involves not only class-level uncertainty but also spatial uncertainty regarding bounding box localization. Moreover, background clutter and visually ambiguous regions can exacerbate aleatoric uncertainty, requiring specialized approaches for reliable estimation.

4.1.4 Challenges and Limitations

Some notable challenges remain in the estimation of aleatoric and epistemic uncertainty:

- **Interaction between the two uncertainties.** In practice, aleatoric and epistemic components are not always cleanly separable, and some estimation methods may underestimate one or both [40].

- **Unreliable aleatoric estimates.** Under challenging conditions, aleatoric uncertainty estimates may become unstable or misleading, undermining confidence calibration [18].

- **Computational cost.** Many approaches, such as ensembles or Monte Carlo dropout, require multiple forward passes, which is expensive for detection models that are already computationally heavy.

- **Dependence on metric and loss design.** The quality of uncertainty estimates is strongly influenced by the choice of uncertainty measure (entropy, variance, mutual information) and how the loss function is structured to capture uncertainty [21].

4.2 Concluding Remarks

Understanding and distinguishing aleatoric and epistemic uncertainties is essential for building reliable ML systems. In object detection tasks for UAV perception, where false positives or overconfidence can compromise safety, uncertainty estimation supports better calibration, improves interpretability, and helps prevent errors in decision-making. While significant progress has been made through ensembles, Bayesian methods, and hybrid approaches, challenges remain in computational efficiency, disentanglement, and robustness. These aspects motivate the use of complementary techniques, which will be discussed in subsequent sections.

4.3 Open-Set Recognition

Traditional classification tasks assume a closed world, where all test samples belong to one of the categories observed during training. In many real applications, however, this assumption fails: test inputs may belong to entirely new and unseen classes. **Open-Set Recognition (OSR)** addresses this challenge by requiring models to not only classify known categories, but also to detect and reject unknowns.

Scheirer et al. [34] first formalized open-set recognition, introducing the notion of *open space risk*, where the decision function must avoid making confident predictions far from the support of known data. Their work proposed compact abating probability models as a principled way to manage this risk. Later, Geng et al. [12] provided a comprehensive survey, highlighting OSR methods ranging from shallow statistical models to deep neural networks, and organizing them into categories such as discriminative approaches, reconstruction-based methods, and generative modeling.

These early contributions illustrate the key principles of OSR: (i) the need for mechanisms to identify unknown inputs during inference, and (ii) the trade-off between recognizing known classes accurately and maintaining caution in unfamiliar regions of feature space. This foundation underpins recent progress across domains such as biometrics, medical imaging, and natural language processing, and motivates the techniques developed in this thesis.

4.4 Spectral Normalization

Training deep networks often leads to the issue that small perturbations in the input can produce disproportionately large changes in the output. This sensitivity is undesirable in tasks that require stability and calibrated predictions, such as uncertainty estimation. A standard way to formalize stability is through the *Lipschitz constant* of a network. A function is said to be K-Lipschitz if its output cannot change faster than K times the change in its input. Bounding the Lipschitz constant therefore ensures smooth behaviour of the model: small variations in the input cannot result in arbitrarily large deviations in the output.

Spectral normalization, introduced by Miyato et al. [28], provides a practical mechanism for constraining the Lipschitz constant of each layer in a neural network. Consider a linear transformation with weight matrix $W \in \mathbb{R}^{m \times n}$. The maximum amplification that this layer can apply to an input vector is given by the operator norm of W, also known as its *spectral norm*:

$$\|W\|_2 = \max_{\|x\|_2=1} \|Wx\|_2.$$

This quantity corresponds to the largest singular value $\sigma_{\max}(W)$ of the matrix. Spectral normalization rescales the weight matrix by its spectral norm, yielding

$$\bar{W} = \frac{W}{\|W\|_2},$$

which guarantees that the linear operator is 1-Lipschitz. Since computing the full singular value decomposition at every update step is computationally prohibitive, in practice the largest singular value is approximated efficiently using a small number of iterations of the power method.

For a network composed of layers $W_1, W_2, \ldots, W_L$ interleaved with activation functions, the Lipschitz constant is bounded above by the product of the individual spectral norms, provided that the activations are themselves 1-Lipschitz (such as ReLU or leaky ReLU with slope ≤ 1):

$$\|f\|_{\text{Lip}} \leq \prod_{\ell=1}^{L} \|W_\ell\|_2.$$

Constraining the spectral norm of each weight matrix therefore provides a global bound on the Lipschitz constant of the network, ensuring that the mapping from input to output is smooth and well-conditioned.

Although originally proposed in the context of stabilizing training of generative adversarial networks, spectral normalization has proved valuable in classification tasks. By bounding the Lipschitz constant of the feature extractor, it produces embeddings that are geometrically well-structured: intra-class features remain compact, while inter-class separability is preserved without excessive distortions. This regularization improves calibration by reducing the tendency of the model to assign high confidence in regions of the input space that were never encountered during training.

This connection is directly relevant for uncertainty estimation. In the work of Mukhoti et al. [29], spectral normalization is applied to the convolutional backbone of a classifier to obtain embeddings suitable for density modeling, such as with Gaussian mixture models. Without spectral normalization, embeddings may become unstable: distances between samples lose meaning, densities collapse, and uncertainty scores fluctuate unpredictably. With spectral normalization, embeddings are regularized, ensuring that uncertainty estimates derived from them are more reliable. In this manner, spectral normalization offers a principled and computationally efficient approach to improving robustness and interpretability in modern classification systems.

In conclusion, spectral normalization can be understood as a mathematical framework for controlling the sensitivity of deep networks. By bounding the singular values of weight matrices, it provides smooth mappings from input to feature space, stabilizes embeddings, and improves the reliability of uncertainty quantification. These properties make spectral normalization a valuable tool not only in generative modeling, but also in classification and uncertainty-aware learning where safety and robustness are of primary importance.

4.5 Gaussian Mixture Models

Gaussian Mixture Models (GMMs) are a classical probabilistic method for modeling complex data distributions. Instead of assuming that all samples come from a single Gaussian distribution, a GMM represents the data as a weighted combination of several Gaussian components, each capturing a different mode or cluster of the data. This makes GMMs particularly well-suited for approximating multimodal distributions.

Formally, a GMM with K components models the probability density of a data point $x \in \mathbb{R}^d$ as

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k),$$

where π_k are the mixture weights with $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$, and $\mathcal{N}(x \mid \mu_k, \Sigma_k)$ denotes a multivariate Gaussian distribution with mean vector $\mu_k \in \mathbb{R}^d$ and covariance matrix $\Sigma_k \in \mathbb{R}^{d \times d}$. Parameters are typically estimated using the Expectation–Maximization (EM) algorithm, which alternates between assigning soft cluster memberships to data points and updating the Gaussian parameters accordingly.

GMMs are widely used for clustering, density estimation, and anomaly detection. In the context of representation learning, they are often applied in the feature space of neural networks to model the distribution of embeddings for each class. This provides a principled way to estimate likelihoods and derive uncertainty scores, which are particularly valuable for tasks such as out-of-distribution detection and open-set recognition.

Chapter 5

Proposed Methodology

5.1 Overview

In this work, we start by enhancing a real-time aerial object detector with per-box **confidence scores indicating whether each detection is out-of-distribution (OOD)**. Our approach is detector-agnostic, requiring only access to feature-space embeddings and thus can be integrated with any modern detector. As illustrated in Figure 5.1, an input image passes through the detector's backbone, which produces a feature representation regularized via spectral normalization to ensure well-behaved embeddings. The transformer-based encoder–decoder then generates object detections, each accompanied by a high-level embedding. These embeddings are fed into **Gaussian Mixture Models (GMMs)**, which estimate per-class likelihoods from which we compute an entropy-based uncertainty score. In parallel, the detector's native softmax confidence is obtained. Both signals are fused during post-processing to prune low-confidence, potentially OOD detections. This post-hoc calibration operates directly on the pretrained backbone without altering the architecture or training process and introduces negligible runtime cost, preserving the detector's real-time throughput.

We then continue by introducing a general, detector-agnostic algorithm that fuses multiple confidence estimates and per-detection features through a lightweight multilayer perceptron (MLP), as illustrated in Fig.5.2a. This formulation provides a flexible framework for improving the area under the ROC curve (AUROC) by learning to combine complementary uncertainty cues. Building on this approach, we propose a model-agnostic embedding-based variant (Fig.5.2b), which leverages the intermediate feature representa-

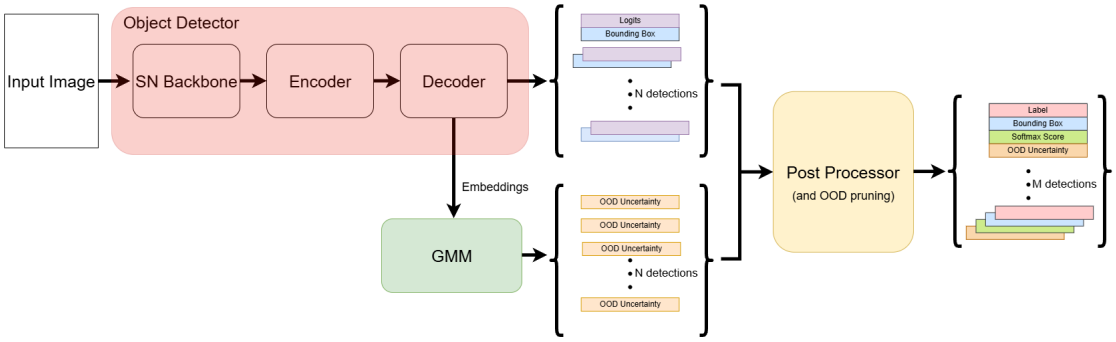
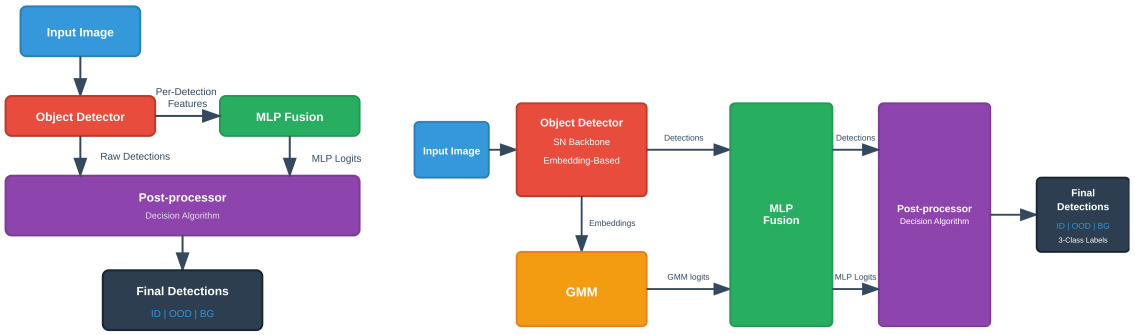


Figure 5.1: Overview of the object detection and uncertainty estimation pipeline.



(a) General architecture (b) Embedding-based architecture

Figure 5.2: Comparison of general and embedding-based feature fusion architectures.

tions produced by modern detectors to achieve enhanced uncertainty calibration. Most importantly, our framework extends beyond standard binary in-/out-of-distribution classification and enables explicit three-class discrimination between in-distribution objects, out-of-distribution objects, and background clutter. This capability is particularly important for reliable autonomous navigation in both terrestrial and aerial environments, where safety-critical decisions must depend on robust uncertainty estimation.

5.2 Uncertainty-Aware Open-Set Detection

5.2.1 Base Detection Framework

Our method is compatible with any modern object detector that produces fixed-dimensional embeddings for each detection. Such detectors typically consist of a backbone network that extracts a feature representation of the input image, followed by an encoder–decoder or head that outputs:

- class logits for category prediction,
- bounding box coordinates, and
- a fixed-dimensional **embedding** capturing high-level appearance information for each detected object.

These embeddings serve as the key input to our density models for estimating semantic uncertainty. To improve feature-space regularity, the convolutional layers in the backbone can optionally be spectrally normalized following [29], enforcing a bi-Lipschitz constraint on the feature mapping. Our method operates post hoc on these embeddings without modifying the detector's architecture, training process, or inference speed.

5.2.2 Feature-Space Density Modeling

Collecting training embeddings

After training, we run the detector on the entire training set. Each prediction is matched to a ground-truth box via the Hungarian assignment built-in into the object detector; the embedding of the matched prediction inherits the ground-truth class label.

This creates a training set where each sample consists of the features (embedding) and the label (class label of the detection) D.

The following matching methodology was also implemented but lead to worse results and thus was abandoned:

- For each image we run it through the detector to get the bounding boxes and the predicted classes
- For each ground truth annotation in the dataset we find the detection with the highest IOU
- If the detection has an IOU of over 50 percent it's considered valid
- For every valid detection we get the embeddings and match to them the ground truth class label of the annotation

Fitting Gaussian mixtures

We continue by training one or multiple GMMs on the previously produced dataset.

- **Single-GMM:** One *full-covariance* Gaussian per class (regularised with a small jitter).
- **Multi-GMM:** A mixture of $K \in \{2, 3, 4\}$ Gaussians per class, fitted with EM.

In the first case one Gaussian of the mixture is responsible for the modeling of the distribution of the embeddings for a signle class. In the second case, each GMM is responsible for the same modeling. Using GMMs per class increases the expressive power of the distribution at the cost of potential overfitting and the approximation of the EM steps.

No OOD data are used at this stage. At inference, each detection embedding is passed through the fitted GMMs to obtain a vector of per-class log-likelihoods; which are subsequently reduced to a single confidence or uncertainty score.

5.2.3 Calibration Techniques

Score pruning

Detections with $S_{\max} < 0.2$ exhibit highly scattered embeddings and dominate AUROC errors (see Fig 5.3). More specifically we observe that the distribution of the id detections has two peaks. The obvious one is the second, centered arround high confidences. The second one, centered around low confidences, accounts for the vast majority of the id detections. At inference the detector produces hundreds of detections, around areas with high density in features. Most of these detections are then pruned through NMS as long as there is one detection of high confidence that shares a high IOU. From the diagram we conclude that ignoring these detections would lead in much better separability of the id to the ood detections. We therefore test every score in a **Raw** setting (no filter) and a **Pruned** settings that discards those low-confidence boxes. Pruning's impact on closed-set mAP is reported in the experiments.

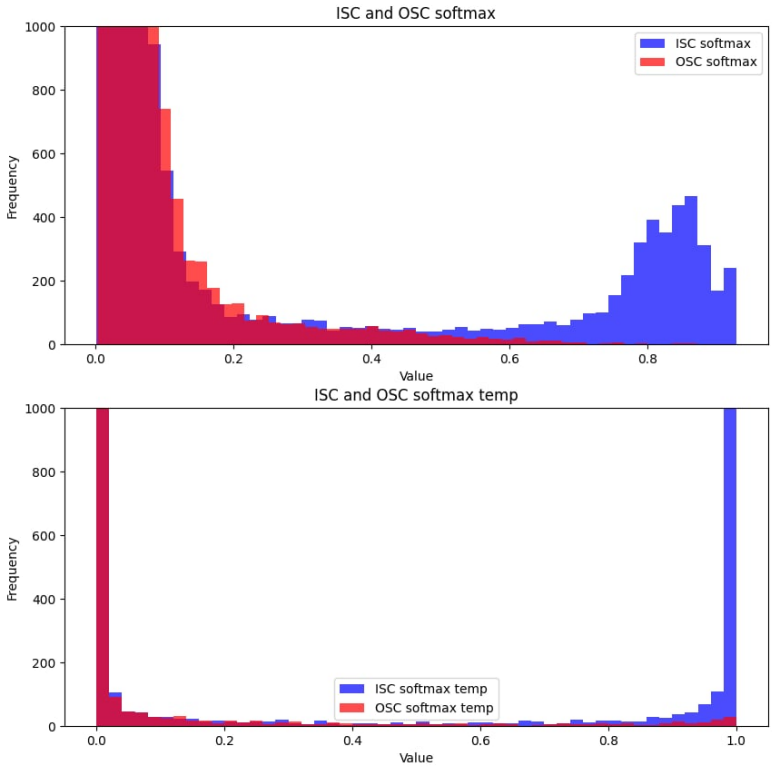


Figure 5.3: Distribution of softmax scores for in-distribution (blue) and out-of-distribution (red) detections. The leftmost peak corresponds to low-confidence detections that are redundant or failed predictions occurring near high-confidence detections. Pruning these low-score detections improves open-set rejection without degrading closed-set mAP, as the correct high-confidence detections remain unaffected.

Temperature scaling

Baseline logits are under-confident, while GMM log-densities can differ by two orders of magnitude, collapsing GMM-derived scores to 0/1. We learn a scalar temperature T_{model} and T_{gmm} on the validation split (negative-log-likelihood minimisation [29]) and rescale both models' densities.

Combining the two toggles (Pruning ×Temperature) yields four evaluation modes per algorithm, model: *Raw, Pruned, Temp, Pruned + Temp*.

5.2.4 Uncertainty Scoring and Ablation Protocol

We begin by describing our main algorithm, which combines sigmoid confidence and GMM-based uncertainty to filter detections. Each detection is assigned both a score and a GMM-derived score (e.g. entropy or density). If both exceed fixed thresholds, the detection is retained; otherwise, it is discarded. The goal is to leverage both complementary signals for improved OOD rejection. We refer to this method as **Joint Thresholding**.

Algorithm 1 Model-Agnostic Open-Set Detection via Joint Thresholding

Definitions:
- Detector output: class logits l, bounding boxes b, embeddings e
- Softmax scores: $p(y|l)$
- GMM entropy: $H_{gmm} = -\sum_y q(y|e) \log q(y|e)$
- Dataset: (X, Y)

procedure TRAIN(X, Y)
  for all images $x \in X$ **do**
    Run detector $\rightarrow$ predictions (b_i, l_i, e_i)
    Match predictions to GT via Hungarian matcher
    Assign e_i to its GT label
  end for
  for all class c with samples $x_c \subset X$ **do**
    $\mu_c \leftarrow \frac{1}{|x_c|} \sum_{x_c} f_\theta(x_c)$
    $\Sigma_c \leftarrow \frac{1}{|x_c|-1} \sum_{x_c} (f_\theta(x_c) - \mu_c)(f_\theta(x_c) - \mu_c)^T$
    $\pi_c \leftarrow \frac{|x_c|}{|X|}$
  end for
end procedure

function OOD_DETECTION$((b, l, e))$
  $p(y) \leftarrow \text{Softmax}(l)$
  $s_{soft} \leftarrow \max_y p(y)$
  $H_{gmm} = -\sum_y q(y|e) \log q(y|e)$
  if $s_{soft} \geq \tau_{soft}$ **and** $H_{gmm} \leq \tau_{gmm}$ **then**
    return ID
  else
    return OOD
  end if
end function

We compare this method against the following standalone confidence scores, each operating on either the logits l (subscripts index classes) or the GMM output:

- **Score confidence**: $\max_c p_c$
- **Softmax density**: $\log \sum_c e^{\ell_c}$
- **Softmax entropy**: $-\sum_c p_c \log p_c$
- **GMM density**: single-Gaussian log-likelihood
- **GMM posterior entropy**: entropy of GMM posteriors
- **Multi-GMM density**: log-likelihood with K Gaussians/class

Algorithm 2 Model-Agnostic Open-Set Detection with MLP Fusion

Definitions:
- *Detector Output*: per detection features f
- *Scores*: confidences and uncertainties derived from features c
- *OOD Dataset*: (X,Y) $\mathcal{D}$

procedure TRAIN_FUSION_MLP($\mathcal{D}, f, c$)
  Select features: From $f \cup c$ select f_s
  Select outputs: choose $K \in \{2, 3\}$
  $S \leftarrow \emptyset$
  for all images $u \in \mathcal{D}_{\text{ID}} \cup \mathcal{D}_{\text{OOD}}$ **do**
    Run detector $\rightarrow$ predictions $x_i = f_s$
    Match with labels for y_i
    $S \leftarrow S \cup \{(x_i, y_i)\}$
  end for
  Train a K-way MLP classifier $g(\cdot)$ on S
  $thresholding(\cdot) \leftarrow$ decision boundaries
end procedure

function CLASSIFY_DETECTION(f_s)
  $logits \leftarrow g(f_s)$
  decision $\leftarrow thresholding(logits)$
  return decision
end function

5.3 Post-Hoc Confidence Fusion with MLP

We first describe the general model-agnostic algorithm for open-set aerial object detection, illustrated in Algorithm 2. The approach constructs a new training set of per-detection features and labels by running a pretrained detector on data containing both in-distribution (ID) and out-of-distribution (OOD) samples. Each detection yields a feature vector that may include raw detector confidences, uncertainty scores, logits, or embeddings, along with a label indicating whether the detection corresponds to an ID object, an OOD object, or background clutter.

Formally, given a detector and a dataset $\mathcal{D} = \mathcal{D}_{\text{ID}} \cup \mathcal{D}_{\text{OOD}}$, the detector is applied to all images. Each prediction is matched to its ground-truth label, producing pairs (X_i, Y_i) where X_i is the feature vector of the detection and $Y_i \in \{\text{ID}, \text{OOD}, \text{BG}\}$ is the class label. This collection of pairs constitutes a new dataset tailored for uncertainty calibration.

From this dataset, a subset of features is selected to serve as input to a lightweight multilayer perceptron (MLP). The desired output configuration is also chosen: a binary classifier ($K = 2$) for ID vs. OOD, or a three-way classifier ($K = 3$) for ID, OOD, and background. The MLP is then trained on the constructed dataset. Lastly, thresholds are tuned on the MLP logits to satisfy desired open-set recognition guarantees (e.g., controlling the false acceptance rate), which need not correspond to a simple arg max decision rule.

At deployment time, for each new detection, the same set of features is extracted, the trained MLP is applied to obtain fused logits, and these are passed through the calibrated decision function. The output is a classification of each detection as ID, OOD, or

background, enabling reliable open-set detection in real time.

5.3.1 Embedding-Based Fusion Algorithm

Building on the general framework described in Section 2, we present a more specific embedding-based implementation tailored to modern detectors that produce per-detection feature embeddings. The methodology follows prior work on embedding-space density modeling [27, 29] and extends it with calibration and pruning strategies, as well as fusion through our MLP.

1. **Detector training with spectral normalization:** The base detector is trained with spectral normalization applied to convolutional layers to enforce bi-Lipschitz continuity and produce well-behaved embeddings.

2. **Temperature calibration of logits:** On a held-out calibration set, scalar temperature parameters are learned to rescale both detector logits and GMM log-likelihoods by minimizing negative log-likelihood. This improves comparability across different uncertainty scores.

3. **Gaussian mixture modeling:** Using the training set, Gaussian mixture models (GMMs) are fitted to the embeddings of each class. Each detection embedding is then mapped to a vector of per-class GMM log-likelihoods, which serve as additional uncertainty signals.

4. **Logit calibration:** The raw GMM log-likelihoods are rescaled using temperature scaling, ensuring that their magnitudes are consistent with detector-derived confidences.

5. **Score pruning:** Detections with low raw confidence scores (sigmoid < 0.2) are discarded, reducing redundancy and eliminating spurious predictions that otherwise dominate AUROC errors.

This procedure provides, for every detection, both calibrated detector scores and GMM-derived logits and confidences. These signals are then used as input features for the fusion MLP described in the previous subsection. The overall embedding-based pipeline is summarized in Algorithm 3, which combines GMM training, fusion MLP training, and the final OOD decision rule.

5.3.2 Detection Classification and Ground Truth Matching

To establish a consistent evaluation framework, we define how detector outputs are categorized relative to ground truth annotations. When comparing detector outputs with ground truth labels, four types of detections emerge:

1. **True Positive ID (TP-ID):** Detections that match with a known ground truth object and predict the correct class label

2. **False Positive ID (FP-ID):** Detections that match with a known ground truth object but predict an incorrect class label

3. **Out-of-Distribution (OOD):** Detections that match with ground truth objects whose class is not present in the training set

4. **Background (BG):** Detections that do not match with any ground truth objects

For the purpose of open-set detection, we classify both TP-ID and FP-ID detections as in-distribution (ID) detections. This design choice separates the problem of ID/ OOD/ background categorization from the problem of correct class prediction within the ID set. This definition differs from some approaches in the literature that consider only correctly classified detections as ID, creating a positive bias in the results. When comparing against prior methods, we recompute their results according to our definition to ensure fair evaluation.

5.3.3 Evaluation Protocol

We evaluate our method under three classification settings: (i) binary ID vs. OOD, (ii) binary ID vs. OOD+background, and (iii) three-class ID vs. OOD vs. background. For all settings, we also track mean average precision (mAP) and frames per second (FPS) to ensure that open-set calibration does not degrade closed-set accuracy or real-time performance.

5.3.4 Domain Shift in MLP Training

Finally, we study the impact of training the fusion MLP with OOD data from sources different from the deployment domain. We find that training on unrelated datasets or synthetic features significantly degrades performance, underscoring the importance of either accessing representative OOD data from the target domain or generating realistic image-domain OOD examples that produce detector features aligned with deployment conditions.

Algorithm 3 Embedding Based Algorithm

Definitions:
- *Detector output*: class logits l, bounding boxes b, embeddings e
- *GMM logits*: ℓ_{gmm}
- *GMM dataset*: (X, Y) with ID class labels for GMMs
- *OOD dataset*: $\mathcal{D} = \mathcal{D}_{\text{ID}} \cup \mathcal{D}_{\text{OOD}}$

procedure TRAIN_GMM(X, Y)
  for all images $x \in X$ **do**
    Run detector $\rightarrow$ predictions (b_i, l_i, e_i)
    Match predictions to GT via Hungarian matcher
    Assign e_i to its GT label
  end for
  for all class c with samples $x_c \subset X$ **do**
    $\mu_c \leftarrow \frac{1}{|x_c|} \sum_{x_c} f_\theta(x_c)$
    $\Sigma_c \leftarrow \frac{1}{|x_c|-1} \sum_{x_c} (f_\theta(x_c) - \mu_c)(f_\theta(x_c) - \mu_c)^T$
    $\pi_c \leftarrow \frac{|x_c|}{|X|}$
  end for
end procedure

procedure TRAIN_FUSION_MLP($\mathcal{D}, f, c$)
  Select features: From $f \cup c$ select f_s
  Select outputs: choose $K \in \{2, 3\}$
  $S \leftarrow \emptyset$
  for all images $u \in \mathcal{D}_{\text{ID}} \cup \mathcal{D}_{\text{OOD}}$ **do**
    Run detector $\rightarrow$ predictions $x_i = f_s$
    Match with labels for y_i
    $S \leftarrow S \cup \{(x_i, y_i)\}$
  end for
  Train a K-way MLP classifier $g(\cdot)$ on S
  thresholding$(\cdot) \leftarrow$ decision boundaries
end procedure

function CLASSIFY_DETECTION(f_s)
  logits $\leftarrow g(f_s)$
  decision $\leftarrow$ *thresholding*(*logits*)
  return decision
end function

Chapter 6

Experimental Setup

In this chapter, we present the design of our experimental evaluation. The experiments are divided into two main parts. The first part focuses on **Joint Thresholding**, where the confidence score derived from the detector and Gaussian mixture models are combined with pruning and temperature scaling. The second part introduces the **Fusion MLP**, a lightweight post-hoc model that learns to combine multiple confidence signals into a single decision rule, extending the problem to a three-class setting. By structuring our experiments in two stages, we highlight the progression from simple threshold-based scoring to more flexible and powerful fusion techniques.

6.1 Common Setup

Before describing each part in detail, we first outline the aspects common to all experiments. All models are trained and evaluated on an NVIDIA A10G-class GPU, while lightweight components such as the fusion MLP are designed to run efficiently on CPU, enabling on-site recalibration if needed.

Across both parts, evaluation follows the same set of metrics. We report the **Area Under the ROC Curve (AUROC)** to measure separability between in-distribution (ID) and out-of-distribution (OOD) detections. We further compute the **True Positive Rate (TPR)** at fixed **Open-Set Recognition (OSR)** levels of 5%, 10%, and 20%, reflecting operational trade-offs between detection and rejection. To ensure closed-set accuracy is not degraded, we measure **mean Average Precision (mAP)** on ID objects under both closed- and open-set conditions. We also track **AUROC**$_{bd}$, where background detections are treated as OOD, reflecting the aerial domain's sensitivity to clutter. For the 3 class setting we also track the **macro pairwise AUROC** to track an average of the separability among classes. Finally, **frames per second (FPS)** are recorded to validate real-time operation.

6.2 Part I: Joint Thresholding

6.2.1 Datasets

The joint thresholding experiments focus on controlled binary open-set detection. We use three sources of data:

- **AOT:** The Airborne Object Tracking (AOT) dataset was introduced as part of the Airborne Object Tracking Challenge, organized by Amazon Prime Air in collaboration with academic partners. It was created to support the development of robust perception systems for unmanned aerial vehicles (UAVs), with the ultimate goal of enabling safer navigation and collision avoidance in shared airspace. We use a subset of it containing the classes of "airplane" and "helicopter", enhanced by an ood class of "drone" for the ablation study. We use a bigger subset, with only in distribution classes to measure the closed-set performance of **Joint Thresholding** when calibrated on domain-shifted open-set conditions.

- **AOT-C:** The AOT-C dataset is a corrupted extension of the Airborne Object Tracking (AOT) benchmark, specifically designed to evaluate the robustness of aerial object detection under real-world conditions. It introduces seven types of synthetic corruptions, grouped into weather effects (fog, rain, low light), sensor noise (ISO noise, color quantization), and defocus blur. Each corruption is applied at four severity levels to ensure controlled evaluation across a range of difficulties. The motivation behind AOT-C is to simulate the environmental and hardware-related challenges encountered in real UAV flights, such as adverse weather or camera noise, while preserving object visibility for fair comparison. We use it to decrease the domain shift between our closed-set training and open-set testing environment.

- **Real Flight Data:** Contains uncontrolled UAV-captured sequences. Here, airplanes and helicopters are considered ID, while drones act as OOD. We opt to use this dataset for our open set evaluation, since its difference in environment, weather conditions, and camera setup create realistic real-world deployment domain shift and test the robustness of our method.

6.2.2 Detector and Variants

The base detector used in our experiments is a transformer-based real-time model (RT-DETR) with a ResNet-50 backbone. We consider two variants: a standard baseline and a spectrally normalized version, in which the convolutional layers are regularized to stabilize the embedding space. In the baseline configuration, the backbone is pretrained and kept frozen during training. This choice is important, as training the backbone from scratch results in significantly worse detection performance due to slow convergence in the early layers of deep networks, a phenomenon commonly linked to vanishing gradients. Moreover, pretraining on large-scale datasets provides richer feature representations and stronger generalization compared to training solely on the specialized aerial dataset. For consistency, both variants employ a pretrained PResNet-50 backbone: the baseline uses the standard ImageNet-1K model, while the spectrally normalized backbone is also pretrained on ImageNet-1K before being frozen for subsequent experiments.

6.2.3 Uncertainty Ablation Study

The first experiment is an ablation study on uncertainty scoring. Each detection is assigned both a logit-derived confidence and a Gaussian mixture–based entropy derived from embeddings. We evaluate several standalone uncertainty scores as well as their combination through joint thresholding. The purpose of this experiment is to determine the most effective scoring method for distinguishing ID and OOD detections. In addition, we test multiple configurations of score pruning and temperature scaling. The goal is not to report

results here, but to establish which combination provides the most reliable foundation for subsequent comparisons.

6.2.4 Closed-Set Accuracy and Runtime

Having established the best configuration, we next study its impact on closed-set detection accuracy and runtime. We measure mAP on ID classes to ensure that open-set filtering does not degrade detection quality, and we record FPS to confirm that real-time performance is preserved. This experiment is critical to show that open-set calibration introduces negligible overhead while maintaining high detection accuracy.

6.2.5 Comparison with Baselines

Finally, we compare joint thresholding against existing uncertainty-based methods, including single-score confidence measures and density modeling approaches. This experiment establishes a clear baseline and highlights whether our thresholding strategy consistently improves AUROC and mAP across datasets. The comparison also illustrates the robustness of joint thresholding under both corrupted and real-world flight conditions.

6.3 Part II: Fusion MLP

6.3.1 Datasets

The second stage expands the evaluation to multi-signal fusion and three-class detection. In addition to AOT-C and real flight data, we also use COCO-OS:

- **AOT-C:** Provides corrupted aerial data for studying robustness.
- **Real Flights:** Tests generalization under natural deployment conditions.
- **COCO-OS:** To evaluate our methods in a more complex and diverse setting, we also make use of the COCO dataset, which is one of the most widely used benchmarks for object detection and contains a large variety of everyday objects in natural scenes. COCO introduces substantial variability in scale, appearance, and background clutter, making it an excellent testbed for open-set evaluation. For our purposes, we construct an open-set variant, COCO-OS, by treating the first 50 categories as in-distribution (ID) classes and the remaining 30 categories as out-of-distribution (OOD). This split allows us to assess the ability of our models to generalize beyond the training distribution while maintaining reliable performance on a challenging large-scale dataset.

6.3.2 Fusion Features and Model

Each detection is represented by a feature vector that includes detector scores (score confidence, entropy, density), GMM-derived scores (likelihoods, entropy), and optionally raw logits or embeddings. These features are input to a lightweight multilayer perceptron (MLP). The MLP is trained for two settings: binary classification (ID vs. OOD) and three-class classification (ID, OOD, background).

6.3.3 Feature Ablation Study

The first experiment evaluates different feature combinations for fusion. We compare detector-only signals, GMM-only signals, and their combination, with and without calibrated logits. This experiment identifies which feature set contributes most to robust uncertainty estimation. We want to avoid feature vectors that lead to low open-set performance, and overfitting. For the second reason we omit the embeddings from the feature space since they are a vector of 256 elements and shatters the OOD datasets, achieving an AUROC of 1, through overfitting.

6.3.4 Two-Class Comparison with Baselines

In the two-class open-set setting (ID vs. OOD), we compare the Fusion MLP against a range of established uncertainty-based methods. These include score thresholding, entropy-based measures, and GMM-derived scores. The purpose of this experiment is to demonstrate that our fusion approach consistently improves AUROC while maintaining the closed-set mAP and the real-time FPS of the base detector. This ensures that the gains in open-set discrimination do not come at the expense of closed-set accuracy or efficiency.

6.3.5 Three-Class Evaluation

We also evaluate our method in the three-class setting, where detections are categorized into ID objects, OOD objects, and background clutter. Here, we compare the Fusion MLP against a double-thresholding baseline, which applies separate thresholds for rejecting background and unknown objects. This experiment tests whether learning a fused decision boundary provides an advantage over handcrafted thresholding, particularly in clutter-heavy aerial imagery where background suppression is crucial.

6.3.6 Domain Shift in Fusion Training

A critical aspect of evaluating the Fusion MLP is understanding how well it generalizes when the out-of-distribution (OOD) data used during training does not match the OOD data encountered at deployment. To study this effect, we design a controlled experiment where the testing OOD data is fixed: in all cases, the evaluation is performed using the real flight dataset, which represents the deployment domain. The variation comes from the source of the OOD data used to train the MLP. Importantly, the MLP does not operate on raw images but only on per-detection feature vectors; thus, the OOD training data is represented entirely within the feature space. We investigate three different strategies for constructing this OOD training set:

1. **Drone Dataset as Proxy OOD.** In the first setting, we select another publicly available UAV dataset, process it through the detector, and use the resulting feature vectors as OOD training samples. The advantage of this approach is that the semantic class of OOD objects (other drones) matches the testing scenario. However, the domains differ significantly: in the proxy dataset, the objects are much larger in the image, often closer to the ground, and recorded under different camera hardware and weather conditions. As a result, the feature distributions of these samples may diverge from those seen in the real flights test set, limiting their utility.

2. **COCO as Proxy OOD.** In the second setting, we sample images from COCO, process them through the detector, and construct OOD training vectors from the resulting detections. This has the positive effect of introducing high variability: COCO covers a wide range of object types and appearances, which helps the MLP establish more general decision boundaries in feature space. On the other hand, the dataset is semantically unrelated to the aerial domain, meaning that the proxy OOD distribution bears little resemblance to the target deployment conditions.

3. **Synthetic OOD in Feature Space.** In the third setting, we generate OOD feature vectors directly, without relying on external images. We first fit distributions to the existing in-distribution (ID) and background detections in feature space. Then, we randomly sample new points and retain only those that fall in regions of very low density relative to the ID and background distributions, effectively placing them in "unoccupied" regions of feature space. While this approach offers flexibility and does not depend on external datasets, it also has limitations: the synthetic data can only be generated from certain feature types, such as logits or GMM logits, which are more tightly connected to the underlying distributions. The resulting OOD samples may therefore fail to fully capture the diversity of real-world OOD conditions.

Through these three complementary settings, we systematically assess the extent to which mismatches between training and testing OOD distributions affect the calibration and robustness of the Fusion MLP. This experiment highlights the trade-offs between semantic similarity, feature diversity, and distributional alignment in open-set training.

6.3.7 Runtime and Model Size Analysis

Finally, we analyze the runtime performance and sensitivity of the MLP to architectural size. By varying the number of hidden units and layers, we test how model complexity influences AUROC, mAP, and FPS. This confirms that even compact MLPs can achieve strong calibration while maintaining real-time throughput.

6.4 Summary

In summary, the experimental setup is designed to progressively evaluate two stages of open-set detection. The joint thresholding experiments validate uncertainty-based filtering in a binary setting, while the fusion MLP experiments extend this to multi-signal fusion and three-class classification. Together, these experiments provide a comprehensive evaluation of both foundational and advanced techniques for robust aerial open-set object detection.

Chapter 7

Results and Analysis

In this chapter we present the results of our experiments. As outlined in the experimental setup, our evaluation proceeds in two stages. The first stage analyzes the performance of the Joint Thresholding method for binary open-set detection, focusing on the effect of uncertainty scoring, calibration, and pruning. The second stage examines the Fusion MLP, comparing it against baselines in the two-class setting and extending to the three-class setting with background separation. Throughout, we evaluate both detection accuracy and open-set robustness, while monitoring real-time performance to ensure practical feasibility.

7.1 Part I: Joint Thresholding

7.1.1 Uncertainty Ablation Study

We begin with an ablation study of uncertainty scoring on the AOT subset enhanced with the drone OOD dataset. Table 7.1 reports AUROC and TPR at fixed OSR levels for different scores. The comparison includes score confidence, softmax entropy, softmax density, GMM density, and GMM posterior entropy, as well as the proposed Joint Thresholding that combines detector and GMM scores. All of the methods are evaluated at both detector variants: Base and Spectrally normalized. Additionally all methods where evaluated with and without pruning the detections scoring under 20% in detector confidence. The results of the unpruned version were uncompetitive across the board, with most of the results scoring AUROC under 0.5 for the reasons described in the Methodology section. For these reasons the results without score pruning are omitted from the table. A notable outlier was the GMM entropy in the following settings:

- **Model Variant:** Base
- **Algorithm:** GMM density
- **GMM temp-scaling:** True
- **Embedding Layer:** 6

For these settings the unpruned method achieved a semi-competitive AUROC of **0.895**. As far as Temperature Scaling is concerned, every algorithm was evaluated with and without temperature scaling applied. To make the table more readable we only incude the best result for each scoring method, indicating whether or not temperature scaling was applied in the last column.

We observe the following:

Table 7.1: AUROC and TPR at fixed OSR levels (5%, 10%, 20%) for each uncertainty scoring method. ✓ indicates that temperature scaling was applied.

Method	AUROC	TPR@5%	TPR@10%	TPR@20%	+Temp
RT-DETR (Base)					
Softmax	0.875	0.506	0.696	0.848	✗
Logsumexp (Density)	0.870	0.536	0.714	0.835	✗
Entropy	**0.939**	**0.810**	**0.873**	**0.913**	✓
GMM Density	0.924	0.783	0.835	0.874	✓
GMM Entropy	0.924	0.725	0.801	0.869	✓
GMM per class	0.927	0.796	0.843	0.887	✓
Joint Thresholding	0.929	0.744	0.829	0.882	✓
RT-DETR + Spectral Normalization					
Softmax	0.916	0.742	0.834	0.884	✓
Logsumexp (Density)	0.870	0.747	0.800	0.837	✓
Entropy	0.939	0.868	0.897	0.911	✓
GMM Density	0.845	0.652	0.707	0.761	✗
GMM Entropy	0.952	0.841	0.906	0.940	✓
GMM per class	0.936	0.712	0.866	0.936	✓
Joint Thresholding	**0.982**	**0.927**	**0.966**	**0.980**	✓

- The best results throughout the study are achived by **Joint Thresholding** using a spectrally normalized backbone. This setting achieves both the highest AUROC and the highest true positive rate at set open set error rates.
- Spectral normalization improves open-set performance on all methods compared to the base variant of the detector.
- Temperature scaling generally improves open-set performance, especially when spectral normalization is also applied.
- **Softmax Entropy** achieves consistently good results in both the base and the spectrally normalized variant; achieving the highest open-set performance in the base variant. This is to be expected since **Softmax Entropy** has been widely used as a metric to measure the sum of **Epistemic** and **Aleatoric Uncertainty**. For these reasons, in the following experiments we also include this method for comparison.

7.1.2 Closed-Set Accuracy and Runtime

Table 7.2 presents the closed-set and open-set mAP ($mAP_{50:95}$) for the evaluated configurations. Here, CS mAP refers to detection performance on the closed-set validation set, which contains only the in-distribution categories (airplanes and helicopters). OS mAP measures performance on the same ID classes in the open-set test set, where unseen drones are also present. Because the closed-set and open-set splits consist of different images, values should not be compared directly across columns, but rather across models and scoring

Table 7.2: Closed-set (CS) and open-set (OS) mAP at IoU 0.5:0.95 (mAP50-95). We report mAP after pruning for each scoring method, using the best configuration per model.

Model	Method	CS mAP	OS mAP
RT-DETR (Base)	Softmax	**54.1**	52.6
	Softmax Density	52.9	53.9
	Entropy	54.0	**55.4**
	GMM Entropy	50.4	52.6
	Joint Thresholding (Ours)	53.7	53.4
RT-DETR + SN	Softmax	**51.9**	56.6
	Softmax Density	49.1	56.6
	Entropy	51.9	56.9
	GMM Entropy	51.7	56.8
	Joint Thresholding (Ours)	51.7	**56.9**

methods. The results indicate that pruning and thresholding do not lead to a significant drop in closed-set performance, and in some cases pruning even improves precision by suppressing low-confidence duplicates. Runtime measurements further confirm that throughput remains above 27 FPS across all configurations, with only negligible overhead from calibration and thresholding.

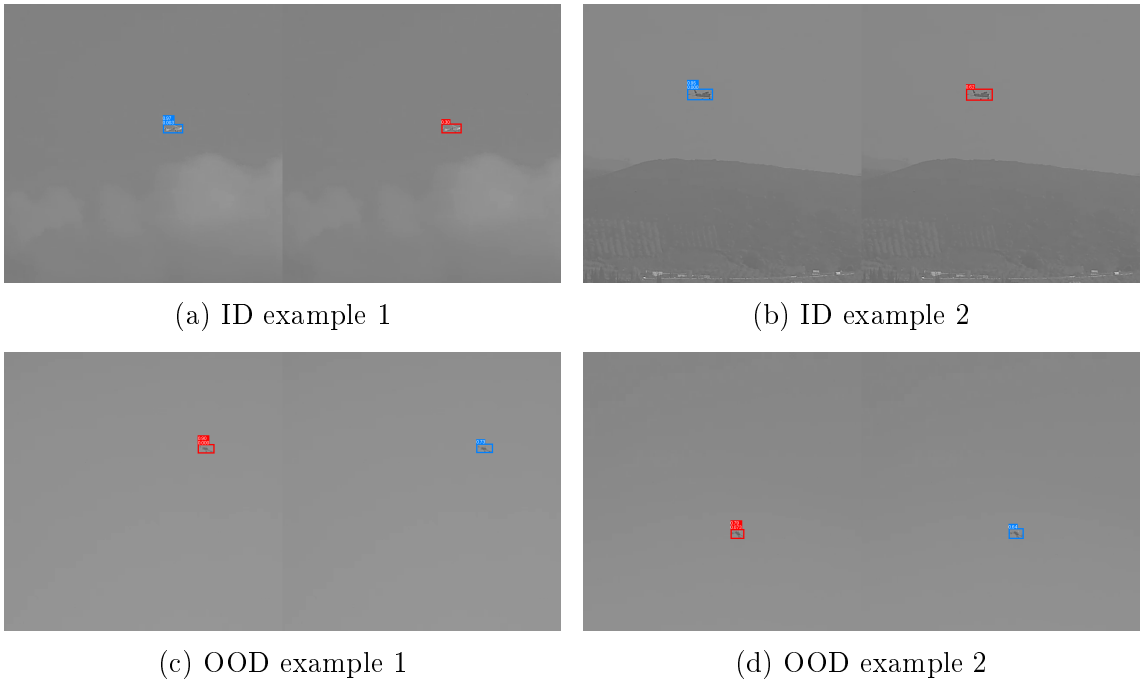
7.1.3 Comparison with Baselines

The most critical evaluation of the Joint Thresholding approach is its ability to generalize beyond synthetic test conditions. To assess this, we train on the AOT-C splits and then evaluate performance on real flight data, treating it as an open-set environment. This setup follows the AOT-C protocol, where synthetic corruptions simulate realistic degradation, and real flights provide a true domain-shift test.

As baselines, we include standard uncertainty-based methods such as softmax entropy and density-based approaches, along with prior open-set detectors including YOLOv5 and GMM-Det. All models are evaluated under the same pipeline for fair comparison. To provide a nuanced view, AUROC is reported under two protocols: one that ignores background detections (AUROC) and one that treats background detections as OOD (AUROC_{bd}). This dual evaluation is particularly important in aerial detection, where background clutter often dominates errors.

Results are summarized in Table 7.3 and the ROC curves are compared in Figure 7.2. We observe that methods relying on a single score, such as softmax entropy or GMM-Det, degrade significantly when applied to real flight conditions. Dynamic lighting, cluttered skies, and sensor noise expose the limitations of calibration alone or of density modeling in isolation. By contrast, Joint Thresholding consistently achieves higher AUROC across both protocols, while maintaining competitive closed-set mAP. The improvement is evident for both the baseline RT-DETR and the spectrally normalized variant, showing that the fusion of detector confidence with embedding-space density modeling provides a more robust rejection mechanism than either source alone.

Beyond numerical results, qualitative inspection (see Fig. 7.1) confirms this trend: Joint Thresholding successfully identifies ID aircraft while rejecting unseen drones as OOD,



(a) ID example 1

(b) ID example 2

(c) OOD example 1

(d) OOD example 2

Figure 7.1: Side-by-side comparison for the *same* image: the **left half of every panel shows RT-DETR (SN)**, the **right half shows YOLO**. **Top row** contains in-distribution (ID) objects, while the **bottom row** contains out-of-distribution (OOD/ID) objects. A **blue** box indicates the detector classified the object as ID; a **red** box indicates the detector judged it OOD. RT-DETR correctly classifies the planes (ID) and the drones (OOD) in all shown cases, whereas YOLO fails on the same images.

avoiding high-confidence false predictions on novel objects. In comparison, YOLO-based baselines often misclassify unknown drones as familiar categories or produce spurious detections with unwarranted confidence. Taken together, these findings highlight the practical advantages of Joint Thresholding: improved robustness under domain shift, more reliable perception, and safer operation of UAVs in real-world environments.

Table 7.3: Performance on real flight data after training on AOT-C. mAP is reported on known classes. AUROC is computed two ways: $\mathbf{AUROC_{bd}}$ treats background detections as OOD; **AUROC** ignores background.

Model	**Method**	**mAP**	$\mathbf{AUROC_{bd}}$	**AUROC**
RT-DETR	Softmax Entropy	40.7	0.837	0.798
RT-DETR	Joint Thresholding	39.3	0.883	0.859
YOLOv5 [3]	Standard	40.0	0.800	0.789
FasterR-CNN	GMM-DET	35.9	0.775	0.723
RT-DETR + SN	**Joint Thresholding**	**41.1**	**0.887**	**0.874**

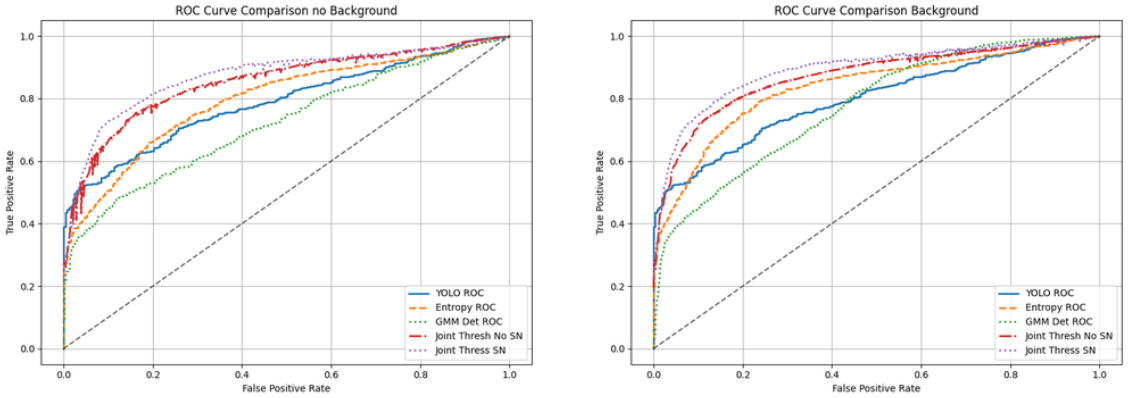


Figure 7.2: Comparison of ROC curves for different methods in open-set real flight data. (a) Results ignoring background detections. (b) Results treating background detections as OOD errors.

7.2 Part II: Fusion MLP

7.2.1 Input Feature Ablation

We first investigate the impact of different input features on the performance of the Fusion MLP. Each detection can be represented by a combination of detector-derived scores (softmax confidence, entropy, density), GMM-derived scores (log-likelihoods, entropy), calibrated logits, and optionally embeddings. Table 7.4 reports AUROC and TPR@OSR for different input configurations. The results highlight several key findings. First, detector-only features provide a reasonable baseline but are limited in capturing the full variability between ID and OOD. Adding GMM-derived scores consistently improves separability, indicating that embedding-space density modeling provides complementary information. Inclusion of calibrated logits further enhances performance by aligning the scale of scores across features. The ablation study confirms that a compact input representation, combining detector scores, GMM signals, and calibrated logits, provides the best balance between performance and generalization.

Table 7.4: Ablation study for MLP inputs in the two-class setting. Each row indicates which input features are included (✓/✗). We report AUROC and TPR at fixed OSR levels (5%, 10%, 20%).

Dataset	Score	Entropy	Density	GMM Entr.	GMM Dens.	Logits	GMM Logits	AUROC	TPR@5%	TPR@10%	TPR@20%
Real Flights	✓	✓	✗	✓	✗	✗	✗	0.891	**0.717**	0.754	0.821
	✓	✓	✓	✓	✓	✗	✗	0.889	0.629	**0.758**	0.819
	✓	✓	✗	✓	✗	✓	✓	0.885	0.681	0.724	0.795
	✓	✓	✓	✓	✓	✓	✓	**0.897**	0.687	0.719	**0.835**
COCO	✓	✓	✗	✓	✗	✗	✗	0.788	0.390	0.493	0.633
	✓	✓	✓	✓	✓	✗	✗	0.788	0.390	0.493	0.633
	✓	✓	✗	✓	✗	✓	✓	**0.894**	**0.636**	**0.739**	0.823
	✓	✓	✓	✓	✓	✓	✓	**0.894**	0.624	0.702	**0.829**

Table 7.5: Comparison of algorithms on Real Flights and COCO datasets. We report mAP, AUROC_{bd}, and AUROC.

Real Flights				
Model	Method	mAP	AUROC_{bd}	AUROC
YOLOv5	Standard	39.3	0.800	0.789
Faster R-CNN	GMM-DET	35.9	0.775	0.723
RT-DETR + SN	Joint	**41.0**	0.887	0.874
RT-DETR + SN	MLP	39.0	**0.887**	**0.897**
COCO				
Model	Method	mAP	AUROC_{bd}	AUROC
YOLOv5	Standard	**44.4**	0.839	0.685
Faster R-CNN	GMM-DET	41.6	0.836	0.872
RT-DETR + SN	Joint	42.0	0.701	0.756
RT-DETR + SN	MLP	42.1	**0.845**	**0.894**

7.2.2 Two-Class Comparison with Baselines

Next, we evaluate the Fusion MLP in the standard two-class open-set setting (ID vs. OOD). Table 7.5 compares our method against baseline scoring approaches, including score thresholding, entropy, and GMM-based methods. We also include Joint Thresholding, our first attempt at combining uncertainty metrics for better open-set performance. The Fusion MLP achieves the highest AUROC across both AOT-C and COCO-OS, while maintaining closed-set mAP at the same level as the base detector.

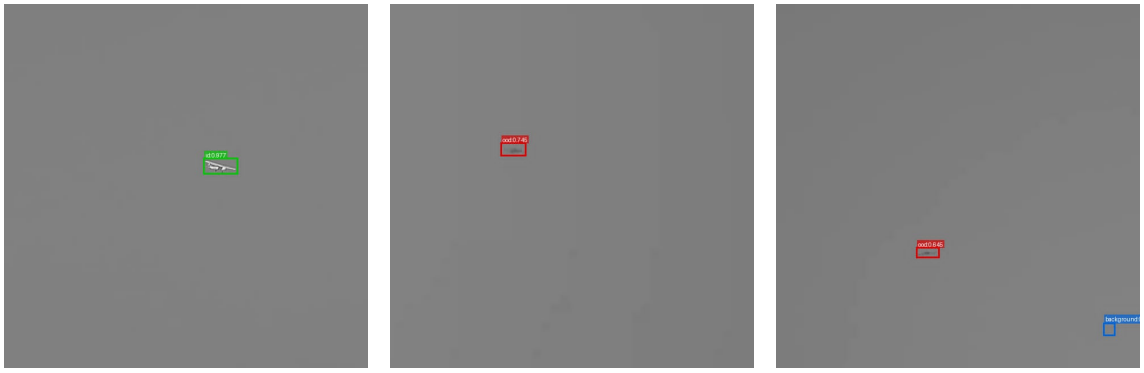
The difference seems at first small, achieving only a moderate 2.7% AUROC improvement from the second best model in each dataset. A closer look at the results reveals the following. The second best method is not the same in both datasets. **Joint Thresholding**, while performing only slightly worse in the Real Flights dataset, underperforms significantly in the more complex COCO-OS dataset. Moreover, the **GMM-Det** algorithm, achieves results really close to the **Fusion MLP** in the COCO-OS dataset, but achieves the worst result out of the four in the real flights dataset. This shows that our model agnostic method is more robust across different domains. Importantly, runtime remains unaffected: throughput stays above 27 FPS, confirming that the additional fusion step introduces negligible computational overhead. These results demonstrate that learning a fused decision boundary from multiple uncertainty signals yields measurable improvements over handcrafted scoring rules, without compromising efficiency or closed-set accuracy.

7.2.3 Three-Class Evaluation

We then extend the evaluation to the three-class setting, where detections are explicitly categorized as ID, OOD, or background. Here, we compare the Fusion MLP against a double-thresholding baseline, which applies independent thresholds for OOD rejection and background suppression. This method is a simple heuristic, basically what open-set methods already do. Our inability to compare with more complex methods comes from

Table 7.6: Three-class results: macro AUROC and Open-Set mAP (higher is better). An asterisk (*) in the result means that all detections in the dataset got pruned.

Algorithm	Real Flights		COCO	
	AUROC	OS mAP	AUROC	OS mAP
Score	0.86	41.5	0.64	36.1
Entropy	0.75	42.7	0.57	39.7
Density	0.79	40.9	0.57	39.0
GMM Entropy	0.78	**45.9**	0.60	32.7
GMM Density	0.81	*	0.46	23.8
MLP	**0.91**	39.1	**0.89**	**41.0**



(a) ID object classification: Airplane (b) OOD object classification: Drone (c) Background Classification

Figure 7.3: Qualitative Results on Real Flights Dataset. ID classifications in green, OOD in red and background in blue. The UAV separates ood objects from background detections improving both safety and efficiency.

the fact that open-set detection has always been framed as a 2 class classification problem until now. Table 7.6 reports results on both COCO-OS and real flight data. The Fusion MLP achieves higher AUROC in both datasets, especially though in the more complex COCO-OS dataset where all other methods significantly underperform. When it comes to open-set mAP, our method remain competitive with others. Qualitative results (Fig. 7.3) demonstrate our findings: the MLP reduces spurious detections in cluttered skies and correctly rejects unseen drones as OOD, outperforming double thresholding in challenging scenarios.

7.2.4 Domain Shift in Fusion Training

We now evaluate how well the Fusion MLP generalizes when trained with OOD data that do not come from the deployment domain, as described in the Experimental Setup.

Training with random detections from COCO achieves the best result with an AUROC of 0.867, benefiting from the diversity of the dataset despite its lack of semantic relation to the aerial domain. Using an unrelated drone dataset yields an AUROC of 0.823, showing that although the OOD class matches semantically, the strong differences in object size,

altitude, and image quality reduce transferability. Synthetic OOD samples generated directly in feature space achieve an AUROC of 0.835, confirming that while this approach avoids dataset mismatch, the resulting points are overly simplistic and fail to capture the richness of real-world OOD conditions. For reference, none of these proxy-based results outperform Joint Thresholding, which remains a competitive baseline.

To further analyze the shortcomings of proxy training, we evaluate the best-performing source (COCO) in the three-class setting. Pairwise AUROCs between ID/OOD, ID/BG, and OOD/BG are reported as (0.722, 0.957, 0.951). These results show that while background can be reliably separated from ID and OOD, distinguishing OOD from ID remains challenging without access to representative OOD training data.

Overall, these experiments highlight the limitations of relying solely on proxy OOD data for calibration. The distribution gap between proxy sources and deployment data is too large for the MLP to generalize effectively. This suggests that future work should focus on generating synthetic image-domain OOD samples that better reflect real-world aerial conditions, thereby narrowing the domain gap and improving the robustness of open-set recognition.

7.2.5 Detection Performance

Finally, we present the closed-set detection performance of our algorithms compared against popular detectors on the AOT and AOT-C datasets. The results can be seen in Table 7.7. Here, we observe the following:

- Detection performance drops significantly when synthetic corruptions are introduced. From this we can conclude, firstly, that training on a harder dataset like the AOT-C, shows the detectors ability to learn beyond optimal conditions. Secondly, it shows that domain shift through adverse weather, or sensor corruptions is an importnat problem that, when not adressed, can compromise UAV safety.
- RT-DETR is the best choice for our experiments. It maintains robust performance on both optimal (AOT) conditions and adverse (AOT-C) conditions, achieving the second best out of the closed-set variant in both datasets.
- Joint Thresholding and 2 class Fusion maintain strong detection performance, comparable with the base RT-DETR variant, while at the same time improving open-set performance.
- 3 class MLP Fusion achieves improved results in detection than the base RT-DETR model. It's detection performance in the clean AOT dataset is comparable to the best detector, while it achieves an **18.0% improvement in mAP** in the corrupted dataset. This improvement leads it to achieve the best results out of all the detectors by a more than 5 point margin. This improvement can be attributed to the rejection of many false positives from the background.

7.3 Summary of Findings

The Fusion MLP experiments provide several key insights:

- A compact input vector combining detector scores, GMM signals, and calibrated logits achieves the best balance between performance and generalization, while high-dimensional embeddings cause overfitting.

Table 7.7: The benchmarking results of 13 object detectors on AOT and AOT-C in terms of Average Precision (AP), inference speed (fps) and model size (M)

Object detector	AP_{clean} ↑	AP_{cor} ↑	fps ↑	Model Size (M) ↓
YOLOv5 [20]	64.6	53.5	99	46.5
YOLOv8 [19]	56.4	41.2	**110**	43.7
YOLOX [11]	**69.3**	43.8	68	54.2
RetinaNet [17, 24]	35.7	20.0	17	**37.9**
FasterR-CNN [31, 36]	52.9	29.7	15	41.3
DiffusionDet [6]	63.8	35.7	30	110.5
DETR [**?**]	58.7	26.1	27	41.2
CenterNet2 [47]	66.2	35.9	24	71.6
GMM-DET (FasterR-CNN) [27]	64.2	48.0	15	41.3
RT-DETR-R50 [45]	66.2	49.6	28	40.1
Joint Thresholding	66.8	49.3	28	40.1
MLP FUSION 2 class	65.0	49.3	27	40.2
MLP FUSION 3 class	69.2	**58.7**	27	40.2

- In the two-class setting, the Fusion MLP consistently surpasses baseline scoring methods, achieving higher AUROC while maintaining closed-set mAP and real-time throughput.
- In the three-class setting, the MLP outperforms the double-thresholding baseline, effectively suppressing background clutter and reducing false positives, subsequently increasing detection performance.
- Domain-shift experiments highlight the importance of training with OOD data that are semantically and distributionally aligned with the deployment environment.
- Runtime analysis confirms that the method remains lightweight, with small MLPs sufficient for strong performance at over 27 FPS.

Together, these findings establish the Fusion MLP as an effective extension of uncertainty-based open-set detection, enabling robust performance in both binary and three-class settings while remaining practical for real-time deployment.

Chapter 8

Discussion

8.1 Interpretation of Key Results

The results of our experiments highlight several important findings regarding the proposed framework for uncertainty-aware open-set detection in UAV settings. A central observation is that combining multiple uncertainty metrics and detector-derived signals encapsulates more information than relying on any single measure. Each type of uncertainty captures a different aspect of the problem: softmax entropy reflects predictive dispersion, density-based scores capture how well features align with known distributions, and calibration reduces systematic confidence misalignment. When combined, these signals provide complementary perspectives on both the epistemic and aleatoric aspects of uncertainty, leading to more robust separation between in-distribution (ID) and out-of-distribution (OOD) samples.

A particularly important dimension of this work is the explicit differentiation between background clutter and OOD targets. In aerial imagery, the majority of the field of view is dominated by background, especially below the horizon where terrain and man-made structures appear. Sensor noise and cluttered environments can generate spurious features that are sufficient to trigger false positives in conventional detectors. By introducing a third class that explicitly models background, the system reduces the risk of conflating background with novel aircraft, thus improving both safety and operational reliability.

Another strength of the approach is its model-agnostic nature. The fusion of signals does not depend on the internal architecture of the underlying detector, which means that the framework can be integrated alongside a wide range of models. This flexibility enables adaptation to different mission profiles or future architectures without requiring significant modifications to the uncertainty estimation or thresholding procedure.

Finally, beyond open-set performance, the framework also improves detection accuracy in challenging conditions. Adverse weather, sensor corruption, and complex backgrounds often degrade baseline detectors by producing high-confidence false alarms. By leveraging joint thresholding and fusion, many of these spurious detections are suppressed. This results in tangible gains in detection performance under real-world conditions, confirming that robustness to background clutter directly translates into better operational reliability.

8.2 Comparison with Related Work

Relative to the existing literature, our framework achieves state-of-the-art performance in the standard two-class open-set detection setting. Across benchmarks, it consistently outperforms baseline scoring approaches such as entropy and density-based metrics, as well as prior open-set detection algorithms like GMM-Det. These results demonstrate the benefits of leveraging complementary signals rather than relying on a single uncertainty measure.

In addition to outperforming prior work in the binary ID/OOD setting, we also introduce a three-class formulation that explicitly separates background from OOD. To our knowledge, this is a novel contribution in the context of open-set object detection. Prior work has typically ignored background, which neglects the unique role of background in aerial imagery and introduces safety risks in UAV deployment. By formalizing background as a separate class, we extend the scope of open-set detection and provide a more principled foundation for robust perception in safety-critical environments.

8.3 Practical Deployment Considerations

From a deployment perspective, the framework satisfies the real-time constraints typical of UAV operation. Inference speed remains well above 27 FPS across all configurations, indicating that the additional fusion and thresholding steps introduce negligible overhead compared to the base detector. This makes the approach suitable for embedded deployment on resource-constrained platforms, where maintaining throughput is critical for safe navigation and timely decision-making.

Another consideration is the role of OOD training data. Our ablation study on proxy OOD sources showed that the choice of OOD data strongly influences performance. While the framework is computationally inexpensive, its success still depends on the availability of representative OOD examples for calibrating thresholds. This is not a limitation unique to our method, but rather a general property of open-set detection systems. Since all threshold-based approaches ultimately require labeled OOD samples for proper calibration, this requirement is consistent with the broader state of the field. Future work may mitigate this dependency through improved synthetic data generation or more principled domain adaptation techniques.

In summary, the framework balances strong open-set robustness with practical deployability. It extends the capabilities of existing methods by combining complementary uncertainties, explicitly modeling background, and maintaining efficiency suitable for UAV integration.

Chapter 9

Conclusion and Future Work

9.1 Conclusion

This thesis has addressed the problem of uncertainty-aware open-set detection in aerial object recognition, with a particular focus on the challenges faced in UAV deployment. We proposed a model-agnostic framework that combines complementary uncertainty signals, introduces a principled distinction between background and out-of-distribution (OOD) samples, and maintains real-time efficiency suitable for embedded systems. Across benchmarks and real-world flight data, the framework consistently improved robustness while preserving competitive closed-set accuracy, demonstrating its practical potential for safety-critical applications.

9.1.1 Summary of Contributions

The main contributions of this work can be summarized as follows:

- We demonstrated that fusing multiple uncertainty measures leads to more reliable separation between ID and OOD detections compared to single-signal baselines.
- We introduced a three-class formulation for open-set detection that explicitly separates background from OOD, addressing a long-standing gap in the literature and reducing the risk of false positives in cluttered environments.
- We established that the framework is model-agnostic, enabling seamless integration with different detectors without architectural modification.
- We confirmed that the proposed methods maintain real-time inference speed, making them practical for UAV deployment where latency and throughput are critical.

9.1.2 Impact on UAV Perception and Safety

By improving robustness under domain shift and reducing high-confidence errors, the proposed framework enhances the reliability of UAV perception systems in complex aerial environments. Differentiating background from OOD and suppressing spurious detections is particularly important for safety, as it reduces the likelihood of false alarms that could trigger unnecessary evasive maneuvers or compromise mission success. Overall, this work contributes toward making UAV object detection systems more dependable and better suited for real-world operation.

9.2 Future Work

Several promising directions remain open for future investigation. First, the Fusion MLP introduced in this thesis can be incorporated into more specialized or custom detectors to assess whether its benefits generalize across architectures and domains. Because the framework is model-agnostic, it can be readily applied alongside different backbones or detection pipelines, potentially uncovering new performance gains.

Second, while this work examined the use of proxy datasets for OOD calibration, the results highlight the importance of having realistic training examples. Future research should explore the generation of simulated OOD data in the image domain, rather than feature space alone. By leveraging modern simulation tools, it may be possible to produce diverse and representative OOD samples that better reflect real-world conditions. Such data would reduce the dependency on domain-specific OOD collections and improve the generalization of open-set detectors.

Taken together, these directions suggest a pathway toward more resilient, generalizable, and practically deployable open-set detection systems for aerial robotics and beyond.

Bibliography

[1] Airborne object tracking dataset. `https://registry.opendata.aws/airborne-object-tracking`. Accessed: 2023-07-23.

[2] Hejer Ammar, Nikita Kiselov, Guillaume Lapouge, and Romaric Audigier. Open-set object detection: towards unified problem formulation and benchmarking. In *European Conference on Computer Vision*, pages 46–61. Springer, 2024.

[3] Anastasios Arsenos, Vasileios Karampinis, Evangelos Petrongonas, Christos Skliros, Dimitrios Kollias, Stefanos Kollias, and Athanasios Voulodimos. Common corruptions for evaluating and enhancing robustness in air-to-air visual object detection. *IEEE Robotics and Automation Letters*, 9(7):6688–6695, 2024.

[4] Anastasios Arsenos, Evangelos Petrongonas, Orfeas Filippopoulos, Christos Skliros, Dimitrios Kollias, and Stefanos Kollias. Nefeli: A deep-learning detection and tracking pipeline for enhancing autonomy in advanced air mobility. *Aerospace Science and Technology*, 155:109613, 2024.

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[6] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19773–19786, 2023.

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] Linhui Dai, Hong Liu, Hao Tang, Zhiwei Wu, and Pinhao Song. Ao2-detr: Arbitrary-oriented object detection transformer, 2022.

[9] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1050–1059, 2016.

[10] Stefano Gasperini, Jan Haug, Mohammad-Ali Nikouei Mahani, Alvaro Marcos-Ramiro, Nassir Navab, Benjamin Busam, and Federico Tombari. Certainnet: Sampling-free uncertainty estimation for object detection. *IEEE Robotics and Automation Letters*, 7(2):698–705, 2021.

[11] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[12] Chuanxing Geng, Sheng-Jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3614–3631, 2020.

[13] Sourish Ghosh, Jay Patrikar, Brady Moon, Milad Moghassem Hamidi, and Sebastian Scherer. Airtrack: Onboard deep learning framework for long-range aircraft detection and tracking. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1277–1283. IEEE, 2023.

[14] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, pages 1321–1330, 2017.

[15] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[16] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[17] Yannick Henon. Pytorch-retinanet: Pytorch implementation of retinanet, 2020.

[18] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

[19] G. Jocher, A. Chaurasia, and J. Qiu. Ultralytics yolo. `https://github.com/ultralytics/ultralytics`, Jan 2023. [Online; accessed August 2, 2025].

[20] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, A. Hogan, lorenzomammana, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, F. Ingham, Frederik, Guilhen, Hatovix, J. Poznanski, J. Fang, L. Yu, changyu98, M. Wang, N. Gupta, O. Akhtar, PetrDvoracek, and P. Rai. ultralytics/yolov5: v3.1 - bug fixes and performance improvements. `https://doi.org/10.5281/zenodo.4154370`, Oct 2020.

[21] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[22] Ruoqi Li, Chongyang Zhang, Hao Zhou, Chao Shi, and Yan Luo. Out-of-distribution identification: Let detector tell which i am not sure. In *European Conference on Computer Vision*, pages 638–654. Springer, 2022.

[23] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[25] Yen-Cheng Liu, Chih-Yao Ma, Xiaoliang Dai, Junjiao Tian, Peter Vajda, Zijian He, and Zsolt Kira. Open-set semi-supervised object detection. In *European conference on computer vision*, pages 143–159. Springer, 2022.

[26] Yufei Liu, Wenqi Wang, Tianfei Zhou, Xiang Li, Jinqing Zhang, and Yi Yang. Modeling aleatoric uncertainty for camouflaged object detection. *IEEE Transactions on Image Processing*, 31:7666–7678, 2022.

[27] Dimity Miller, Niko Sünderhauf, Michael Milford, and Feras Dayoub. Uncertainty for identifying open-set errors in visual object detection. *IEEE Robotics and Automation Letters*, 7(1):215–222, 2021.

[28] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[29] Jishnu Mukhoti, Andreas Kirsch, Joost Van Amersfoort, Philip HS Torr, and Yarin Gal. Deep deterministic uncertainty: A new simple baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24384–24394, 2023.

[30] Roberto Opromolla and Giancarmine Fasano. Visual-based obstacle detection and tracking, and conflict detection for small uas sense and avoid. *Aerospace Science and Technology*, 119:107167, 2021.

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.

[32] Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, et al. Grounding dino 1.5: Advance the" edge" of open-set object detection. *arXiv preprint arXiv:2405.10300*, 2024.

[33] Douglas Reynolds. Gaussian mixture models. In *Encyclopedia of biometrics*, pages 827–832. Springer, 2015.

[34] Walter J Scheirer, Anderson de Rezende Rocha, Anil Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, 2013.

[35] Walter J. Scheirer, Anderson Rocha, Archana Sapkota, and Terrance E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, 2013.

[36] sovit-123. Faster r-cnn pytorch training pipeline, 2025.

[37] Binyi Su, Hua Zhang, Jingzhi Li, and Zhong Zhou. Toward generalized few-shot open-set object detection. *IEEE Transactions on Image Processing*, 33:1389–1402, 2024.

[38] Chen Sun, Ruihe Zhang, Yukun Lu, Yaodong Cui, Zejian Deng, Dongpu Cao, and Amir Khajepour. Toward ensuring safety for autonomous driving perception: Standardization progress, research advances, and perspectives. *IEEE Transactions on Intelligent Transportation Systems*, 25(5):3286–3304, 2023.

[39] Xian Sun, Peijin Wang, Zhiyuan Yan, Feng Xu, Ruiping Wang, Wenhui Diao, Jin Chen, Jihao Li, Yingchao Feng, Tao Xu, Martin Weinmann, Stefan Hinz, Cheng Wang, and Kun Fu. Fair1m: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery, 2021.

[40] Matias Valdenegro-Toro and Javier Saromo. Uncertainty estimation and its applications in deep learning. *SN Computer Science*, 3(3):1–13, 2022.

[41] Ke Wang, Chongqiang Shen, Xingcan Li, and Jianbo Lu. Uncertainty quantification for safe and reliable autonomous vehicles: A review of methods and applications. *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[42] Samuel Wilson, Tobias Fischer, Niko Sünderhauf, and Feras Dayoub. Hyperdimensional feature fusion for out-of-distribution detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2644–2654, 2023.

[43] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images, 2019.

[44] Yifei Yang, Zhongxiang Zhou, Jun Wu, Yue Wang, and Rong Xiong. Class semantics modulation for open-set instance segmentation. *IEEE Robotics and Automation Letters*, 9(3):2240–2247, 2024.

[45] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. DETRs Beat YOLOs on Real-time Object Detection, June 2024.

[46] Ye Zheng, Zhang Chen, Dailin Lv, Zhixing Li, Zhenzhong Lan, and Shiyu Zhao. Air-to-air visual detection of micro-uavs: An experimental evaluation of deep learning. *IEEE Robotics and Automation Letters*, 6(2):1020–1027, 2021.

[47] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. In *arXiv preprint arXiv:2103.07461*, 2021.

[48] Zhongxiang Zhou, Yifei Yang, Yue Wang, and Rong Xiong. Open-set object detection using classification-free object proposal and instance-level contrastive learning. *IEEE Robotics and Automation Letters*, 8(3):1691–1698, 2023.