



ΔΙΔΡΥΜΑΤΙΚΟ ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ  
ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
ΣΤΡΑΤΙΩΤΙΚΗΣ ΣΧΟΛΗΣ ΕΥΕΛΠΙΔΩΝ ΚΑΙ  
ΣΧΟΛΗΣ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ  
ΔΙΟΙΚΗΣΗΣ  
ΤΟΥ ΠΟΛΥΤΕΧΝΕΙΟΥ ΚΡΗΤΗΣ



Διπλωματική Εργασία

**Αναγνώριση Κεραυνών σε Καταιγίδα με τη χρήση  
Τεχνητής Νοημοσύνης**

Επιβλέπων Καθηγητής: Ν. Παπαδάκης

Παππά Μαρία Ιωάννα

ΙΩΑΝΝΙΝΑ

2025



INTER-INSTITUTIONAL AND INTER-  
DEPARTMENT PROGRAM OF POSTGRADUATE  
STUDIES OF THE HELLENIC MILITARY  
ACADEMY AND THE SCHOOL OF  
PRODUCTION AND MANAGEMENT OF THE  
TECHICAL UNIVERSITY OF CRETE



Master Thesis

**Lightning Detection in Thunderstorms Using Artificial  
Intelligence**

Supervisor professor: N. Papadakis

Pappa Maria Ioanna

IOANNINA

2025

Η Τριμελής Εξεταστική Επιτροπή

Ο Επιβλέπων Καθηγητής

Νικόλαος Παπαδάκης

Βασίλειος Κουικόγλου

Αντώνιος Λίτκε

Νικόλαος Παπαδάκης



Αναπληρωτής Καθηγητής

Καθηγητής

Λέκτορας

## ΠΕΡΙΛΗΨΗ

Η αναγνώριση κεραυνών αποτελεί ένα κρίσιμο πεδίο έρευνας με ποικίλες εφαρμογές, καθώς ο κεραυνός χαρακτηρίζεται ως ένα επικίνδυνο φυσικό φαινόμενο με σοβαρές επιπτώσεις για την ανθρώπινη ζωή, τις υποδομές αλλά και το περιβάλλον. Στην παρούσα εργασία εξετάζεται η χρήση τεχνητής νοημοσύνης για την αναγνώριση κεραυνών σε καταιγίδες.

Η βιβλιογραφική ανασκόπηση εξετάζει τις θεωρητικές προσεγγίσεις και τις απαραίτητες έννοιες που σχετίζονται με το θέμα αυτό. Αναφέρονται οι παραδοσιακές μέθοδοι στην αναγνώριση κεραυνών, οι οποίες βασίζονται κυρίως στις αρχές της φυσικής και των μαθηματικών. Επιπλέον, αναφέρεται η χρήση της τεχνητής νοημοσύνης, των νευρωνικών δικτύων και διεργασίες που αφορούν την επεξεργασία της εικόνας και την υπολογιστική όραση.

Για την υλοποίηση του πειραματικού μέρους, αναλύθηκαν και αναπτύχθηκαν τα μοντέλα βαθιάς μάθησης YOLOv5 και YOLOv8, τα οποία εκπαιδεύτηκαν σε ένα σύνολο δεδομένων από εικόνες, οι οποίες συλλέχθηκαν από διάφορες πηγές. Τα YOLO αποτελούν δημοφιλή μοντέλα βαθιάς μάθησης για την ανίχνευση αντικειμένων σε πραγματικό χρόνο, ενώ θεωρούνται μοντέλα State-of-The-Art (SOTA) σε πολλές εφαρμογές. Η εργασία επικεντρώνεται στην ανάλυση της απόδοσης των μοντέλων αυτών στην αναγνώριση κεραυνών.

Τέλος, πραγματοποιείται λεπτομερής εξέταση των πειραματικών αποτελεσμάτων και αποδεικνύονται οι δυνατότητες της τεχνητής νοημοσύνης στην ανίχνευση κεραυνών.

**Λέξεις κλειδιά:** Αναγνώριση Κεραυνών, Τεχνητή Νοημοσύνη, Νευρωνικά Δίκτυα, Επεξεργασία Εικόνας, YOLOv5, YOLOv8

# ABSTRACT

Lightning detection is a critical field of research with diverse applications, as lightning is characterized as a dangerous natural phenomenon with serious consequences for human life, infrastructure and the environment. This master thesis examines the use of artificial intelligence for lightning detection in thunderstorms.

The literature review examines the theoretical approaches and necessary concepts related to the subject. Traditional methods for lightning detection are mentioned, which are mainly based on the principles of physics and mathematics. In addition, the use of artificial intelligence, neural networks and processes related to image processing and computer vision are mentioned.

For the implementation of the experimental part, the deep learning models YOLOv5 and YOLOv8, were analyzed and developed, which were trained on a dataset of images collected from various sources. YOLOs are popular deep learning models for real-time object detection and are considered a State-of-The-Art (SOTA) model in many applications. The work focuses on analyzing the performance of these models in lightning detection.

Finally, a detailed examination of the experimental results is carried out and the capabilities of artificial intelligence in lightning detection are demonstrated.

**Keywords:** Lightning Detection, Artificial Intelligence, Neural Networks, Image Processing, YOLOv5, YOLOv8

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Με την ολοκλήρωση της διπλωματικής μου εργασίας, θεωρώ απαραίτητο να εκφράσω τις ευχαριστίες μου στα άτομα που συντέλεσαν και συνέφεραν σημαντικά στην περάτωση της.

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Παπαδάκη Νικόλαο για το ενδιαφέρον, τη καθοδήγηση αλλά και τις χρήσιμες συμβουλές που μου προσέφερε.

Ακόμη, θα ήθελα να ευχαριστήσω τους κοντινούς μου ανθρώπους, οικογένεια και φίλους, για την υποστήριξη καθ' όλη τη διάρκεια της ακαδημαϊκής μου πορείας.

# ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	iv
ABSTRACT .....	v
ΕΥΧΑΡΙΣΤΙΕΣ.....	vi
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ, ΠΙΝΑΚΩΝ.....	ix
ΕΙΣΑΓΩΓΗ.....	1
1. Οι κεραυνοί ως φυσικά φαινόμενα .....	3
1.1. Κεραυνοί .....	3
1.2. Δημιουργία κεραυνών .....	4
1.3. Τύποι κεραυνών.....	4
1.3.1. Μέσα στο ίδιο το νέφος.....	5
1.3.2. Μεταξύ διαφορετικών νεφών.....	6
1.3.3. Μεταξύ του νέφους και του εδάφους.....	6
1.4. Το φαινόμενο της βροντής.....	8
1.5. Σημείο κρούσης κεραυνού.....	8
1.6. Κίνδυνοι και Επιπτώσεις .....	9
1.7. Εφαρμογές αναγνώρισης κεραυνών .....	10
1.8. Προστασία από κεραυνούς.....	12
1.9. Μέθοδοι ανίχνευσης κεραυνών.....	12
1.9.1. Ανίχνευση μέσω ηλεκτρομαγνητικών κυμάτων.....	13
1.9.1.1. Το δίκτυο EUCLID.....	13
1.9.1.2. Το σύστημα ZEYΣ .....	14
1.9.2. Ακουστικά κύματα .....	15
1.9.3. Οπτικά σήματα .....	15
2. Παραδοσιακές μέθοδοι αναγνώρισης κεραυνών.....	17
2.1. Μέθοδοι με fractals .....	17
2.2. Ντετερμινιστικοί μαθηματικοί αλγόριθμοι .....	18
2.3. Μέθοδοι βασισμένες σε διαφορικές εξισώσεις.....	18
2.3.1. Αριθμητική ανάλυση και πεπερασμένα στοιχεία.....	19
2.4. Αλγόριθμοι και υπολογιστικές μέθοδοι.....	19
2.4.1. Fast Fourier Transform (FFT).....	19
2.4.2. Monte Carlo.....	20
3. Η χρήση της Τεχνητής Νοημοσύνης στην ανίχνευση κεραυνών .....	21
3.1. Επεξεργασία εικόνας .....	21
3.2. Επεξεργασία εικόνας με τη χρήση Τεχνητής Νοημοσύνης .....	23
3.3. Επεξεργασία εικόνας από υπολογιστή.....	25
3.4. Τεχνητά Νευρωνικά Δίκτυα.....	26
3.5. Δομή Τεχνητών Νευρωνικών Δικτύων .....	27
3.5.1. Λειτουργία του Νευρώνα.....	27

3.5.2.	Εκπαίδευση Τεχνητών Νευρωνικών Δικτύων.....	28
3.6.	Συνελικτικά Νευρωνικά Δίκτυα .....	29
3.6.1.	Αρχιτεκτονική Συνελικτικών Νευρωνικών Δικτύων .....	29
3.6.2.	Λειτουργία Συνελικτικών Νευρωνικών Δικτύων.....	30
3.7.	Ανίχνευση αντικειμένων με τα Συνελικτικά Νευρωνικά Δίκτυα .....	31
4.	Μεθοδολογία.....	33
4.1.	Συλλογή δεδομένων.....	33
4.1.1.	Πηγή δεδομένων .....	34
4.1.2.	Δομή του Dataset.....	34
4.2.	Επεξεργασία δεδομένων .....	35
4.3.	Επεξεργασία εικόνων και υπολογιστική όραση.....	38
4.4.	YOLOv5 .....	39
4.4.1.	YOLOv5: Backbone.....	41
4.4.2.	YOLOv5: Neck.....	42
4.4.3.	YOLOv5: Head.....	42
4.4.4.	Αρχιτεκτονική του YOLOv5 .....	43
4.5.	Δημιουργία του μοντέλου αναγνώρισης κεραυνών στο YoloV5 .....	44
4.6.	YOLOv8.....	45
4.7.	Κατηγοριοποίηση των τύπων των μοντέλων YOLO .....	47
4.8.	Google Colab.....	47
4.9.	Αξιολόγηση μοντέλων στην αναγνώριση κεραυνών .....	48
4.9.1.	Ανάλυση του γραφήματος απόδοσης YOLO μοντέλων.....	49
5.	Υλοποίηση μοντέλου .....	51
5.1.	Προετοιμασία δεδομένων για το YOLOv5 .....	54
5.2.	Εκπαίδευση του μοντέλου .....	54
5.2.1.	YOLOv5 .....	54
5.2.2.	Ο κώδικας του train().....	57
5.2.3.	YOLOv8 .....	59
5.3.	Αξιολόγηση μοντέλου .....	63
5.3.1.	Απώλειες (Losses) .....	63
5.3.2.	Μετρικές.....	63
6.	Σύγκριση πειραματικών μοντέλων.....	65
6.1.	1 <sup>ο</sup> πειραματικό μοντέλο.....	65
6.1.1.	YOLOv5s.pt.....	65
6.1.1.1.	Train batches.....	66
6.1.1.2.	Labels .....	66
6.1.1.3.	Results .....	68
6.1.1.4.	Validation .....	69
6.1.2.	YOLOv5m.pt.....	74



6.1.2.1.	Train batches.....	74
6.1.2.2.	Labels .....	74
6.1.2.3.	Results .....	75
6.1.2.4.	Validation .....	76
6.2.	Το 2 <sup>ο</sup> πειραματικό μοντέλο.....	80
6.2.1.	yolov8s.pt.....	80
6.2.1.1.	Train batches.....	80
6.2.1.2.	Labels .....	80
6.2.1.3.	Confusion matrices .....	81
6.2.1.4.	Results .....	82
6.2.1.5.	Validation .....	83
6.2.2.	Yolov8m.pt .....	87
6.2.2.1.	Train batches.....	87
6.2.2.2.	Labels .....	87
6.2.2.3.	Confusion matrices .....	88
6.2.2.3.	Results .....	88
6.2.2.4.	Validation .....	89
7.	Συμπεράσματα .....	93
	ΒΙΒΛΙΟΓΡΑΦΙΑ .....	95
	ΠΑΡΑΡΤΗΜΑ .....	101

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ, ΠΙΝΑΚΩΝ

Σχήμα 1: Κεραυνοί (NASA) .....	4
Σχήμα 2: Τα φορτία ενός νέφους κατά τη διάρκεια καταιγίδας (NOAA) .....	6
Σχήμα 3: Τα στάδια δημιουργίας ενός κεραυνού μεταξύ νέφους και εδάφους (NOAA) .....	7
Σχήμα 4: Κρούση κεραυνού σε κορμό δέντρου (Monster Tree Service) .....	9
Σχήμα 5: Κρούση κεραυνού σε δέντρο (panoramatreeservice).....	10
Σχήμα 6: Χάρτης κάλυψης περιοχών του δικτύου EUCLID (EUCLID) .....	13
Σχήμα 7: Το σύστημα ZEYΣ (METEO) .....	14
Σχήμα 8: : Τα βασικά χρώματα μίας έγχρωμης εικόνας (medium) .....	22
Σχήμα 9: Δημιουργία ψηφιακής εικόνας (legendtechz) .....	24
Σχήμα 10: Αποτέλεσμα της δειγματοληψίας και της κβαντοποίησης της εικόνας (legendtechz) .	25
Σχήμα 11: Δομή Τεχνητών Νευρωνικών Δικτύων (medium) .....	26
Σχήμα 12: Μοντέλο Τεχνητού Νευρώνα (Xu et al., 2024) .....	28
Σχήμα 13: Συνελκτικά Νευρωνικά Δίκτυα ( Viejo, 2020).....	31
Σχήμα 14: Εικόνα από το Dataset_Training.....	34
Σχήμα 15: Εικόνα που χρησιμοποιήθηκε ως αρνητικό παράδειγμα για την επαλήθευση του μοντέλου .....	35
Σχήμα 16: Ορισμός ετικετών στην εικόνα από το εργαλείο Labelling .....	37
Σχήμα 17: Τα pixels εικόνας του dataset (Δημιουργήθηκε με τη χρήση τεχνητής νοημοσύνης με το εργαλείο δημιουργίας εικόνων DALL·E).....	38
Σχήμα 18: Το inference flowchart του YOLOv5 (Tsang, 2023) .....	41
Σχήμα 19: Η δομή δικτύου του YOLOv5 (Tsang, 2023) .....	41
Σχήμα 20: YOLOv5: Neck (Tsang, 2023).....	42
Σχήμα 21: YOLOv5: Head (Tsang, 2023).....	43
Σχήμα 22: Λεπτομερής αρχιτεκτονική του YOLOv5 (Tsang, 2023) .....	44
Σχήμα 23: Η αρχιτεκτονική του YOLOv8 (Timilsina, 2024) .....	45
Σχήμα 24: Επιδόσεις των μοντέλων YOLOv8 και YOLOv10 (ultralytics) .....	46
Σχήμα 25: Επιδόσεις των μοντέλων YOLO (ultralytics) .....	49
Σχήμα 26: Σύγκριση μοντέλων σε latency-accuracy (Αριστερά) and FLOPs-accuracy (Δεξιά) trade-offs (github) .....	49
Σχήμα 27: Το Dataset_Training του φακέλου Dataset_Lightning_Detection.....	51
Σχήμα 28: Το Dataset_Testing του φακέλου Dataset_Lightning_Detection.....	52
Σχήμα 29: Εικόνα από το Dataset_Testing .....	53
Σχήμα 30: Το αρχείο custom_data .....	55
Σχήμα 31: Η εκπαίδευση του μοντέλου YOLOv5 .....	56
Σχήμα 32: Ο κώδικας του train() .....	59
Σχήμα 33: Η εντολή import os.....	60
Σχήμα 34: Η εγκατάσταση του YOLOv8 .....	60
Σχήμα 35: Η εκπαίδευση του YOLOv8 .....	61
Σχήμα 36: Η παράμετροι εκπαίδευσης του YOLOv8.....	62
Σχήμα 37: Train batch 0 και 1 για το yolov5s.pt.....	66
Σχήμα 38: Labels του yolov5s.pt .....	67
Σχήμα 39: Τα αποτελέσματα της εκπαίδευσης του yolov5s.pt .....	68
Σχήμα 40: Σετ επικύρωσης 0 για το yolov5s.pt .....	70
Σχήμα 41: Προβλεψεις του σετ επικύρωσης 0 για το yolov5s.pt.....	71
Σχήμα 42: Σετ επικύρωσης 1 για το yolov5s.pt .....	72
Σχήμα 43: Προβλεψεις του σετ επικύρωσης 1 για το yolov5s.pt.....	73
Σχήμα 44: Train batch 0 και 1 για το yolov5m.pt .....	74
Σχήμα 45: Labels του yolov5m.pt.....	74

Σχήμα 46: Τα αποτελέσματα της εκπαίδευσης του yolov5m.pt.....	75
Σχήμα 47: Σετ επικύρωσης 0 για το yolov5m.pt.....	76
Σχήμα 48: Προβλεψεις του σετ επικύρωσης 0 για το yolov5m.pt.....	77
Σχήμα 49: Σετ επικύρωσης 1 για το yolov5m.pt.....	78
Σχήμα 50: Προβλεψεις του σετ επικύρωσης 1 για το yolov5m.pt.....	79
Σχήμα 51: Train batch 0 και 1 για το yolov8s.pt.....	80
Σχήμα 52: Labels του yolov8s.pt.....	80
Σχήμα 53: Confusion matrices του yolov8s.pt.....	81
Σχήμα 54: Τα αποτελέσματα της εκπαίδευσης του yolov8s.pt.....	82
Σχήμα 55: Σετ επικύρωσης 0 για το yolov8s.pt.....	83
Σχήμα 56: Προβλεψεις του σετ επικύρωσης 0 για το yolov8s.pt.....	84
Σχήμα 57: Σετ επικύρωσης 1 για το yolov8s.pt.....	85
Σχήμα 58: Προβλεψεις του σετ επικύρωσης 1 για το yolov8s.pt.....	86
Σχήμα 59: Train batch 0 και 1 για το yolov8m.pt.....	87
Σχήμα 60: Labels του yolov8m.pt.....	87
Σχήμα 61: Confusion matrices του yolov8m.pt.....	88
Σχήμα 62: Τα αποτελέσματα της εκπαίδευσης του yolov8m.pt.....	88
Σχήμα 63: Σετ επικύρωσης 0 για το yolov8m.pt.....	89
Σχήμα 64: Προβλεψεις του σετ επικύρωσης 0 για το yolov8m.pt.....	90
Σχήμα 65: Σετ επικύρωσης 1 για το yolov8m.pt.....	91
Σχήμα 66: Προβλεψεις του σετ επικύρωσης 1 για το yolov8m.pt.....	92

Πίνακας 1: Παράμετροι εκπαίδευσης.....	65
--	----

# ΕΙΣΑΓΩΓΗ

Οι κεραυνοί αποτελούν ένα φυσικό φαινόμενο, το οποίο παρατηρείται συχνά στη φύση και μπορεί να επιφέρει σοβαρές επιπτώσεις για την ανθρώπινη ζωή, τις υποδομές αλλά και το φυσικό το περιβάλλον. Εμφανίζονται κυρίως συνοδευόμενοι από καταιγίδες και η ανίχνευση τους θεωρείται κρίσιμη. Η αναγνώριση των κεραυνών με τη χρήση της τεχνητής νοημοσύνης αποτελεί ένα σημαντικό και καινοτόμο πεδίο έρευνας με ποικίλες εφαρμογές.

Αντικείμενο της παρούσας διπλωματικής εργασίας, αποτελεί η ανάπτυξη ενός μοντέλου αναγνώρισης κεραυνών σε καταιγίδες με τη χρήση τεχνητής νοημοσύνης.

Το πρώτο κεφάλαιο περιλαμβάνει τη βιβλιογραφική ανασκόπηση, γίνεται εξέταση των κεραυνών ως φυσικό φαινόμενο, καθώς και όλων εκείνων των απαραίτητων εννοιών για τη καλύτερη κατανόηση του φαινομένου αυτού. Αναλύεται ο τρόπος δημιουργία τους, οι τύποι στους οποίους διακρίνονται, καθώς και οι κίνδυνοι και οι επιπτώσεις που προκύπτουν από την εμφάνισή τους. Επιπλέον, αναφέρονται οι εφαρμογές που βρίσκει η αναγνώριση κεραυνών, αλλά και μέθοδοι προστασίας από αυτούς.

Το δεύτερο κεφάλαιο, επικεντρώνεται στις παραδοσιακές μεθόδους αναγνώρισης κεραυνών, χρησιμοποιώντας ντετερμινιστικούς μαθηματικούς αλγόριθμους, διαφορικές εξισώσεις, αριθμητική ανάλυση και τεχνικές υπολογιστικών μεθόδων, οι οποίες χρησιμοποιούνται για την καταγραφή και ανάλυση των φαινομένων αυτών. Στο τρίτο κεφάλαιο, εξετάζεται η χρήση της τεχνητής νοημοσύνης στην επεξεργασία εικόνας, οι έννοιες των Νευρωνικών Δικτύων, αλλά και συγκεκριμένα η ανίχνευση αντικειμένων με τα Συνελικτικά Νευρωνικά Δίκτυα.

Στο τέταρτο κεφάλαιο, αναλύεται η μεθοδολογία που ακολουθήθηκε για τη συλλογή και επεξεργασία δεδομένων, ενώ παρουσιάζονται τα μοντέλα βαθιάς μάθησης YOLO που θα χρησιμοποιηθούν για την ανάπτυξη μοντέλων αναγνώρισης κεραυνών. Η αρχιτεκτονική αυτών των μοντέλων και οι τεχνικές εκπαίδευσης αναφέρονται αναλυτικά, όπως και η διαδικασία αξιολόγησής τους.

Στο πέμπτο κεφάλαιο, πραγματοποιείται η υλοποίηση τους, ενώ στο έκτο κεφάλαιο, γίνεται η σύγκριση των πειραματικών μοντέλων που αναπτύχθηκαν,

εστιάζοντας στην αξιολόγηση της απόδοσης και στην ανάλυση των αποτελεσμάτων.

Τέλος, η εργασία ολοκληρώνεται με τα συμπεράσματα του έβδομου κεφαλαίου, τα οποία προκύπτουν από την εφαρμογή των παραπάνω μεθόδων στην αναγνώριση κεραυνών.

# 1. Οι κεραυνοί ως φυσικά φαινόμενα

Μία φυσική καταστροφή αποτελεί ένα φυσικό γεγονός, το οποίο επηρεάζει τόσο την ανθρώπινη ζωή και δραστηριότητα όσο και το φυσικό περιβάλλον. Οι κεραυνοί ανήκουν σε αυτές τις φυσικές καταστροφές και μάλιστα επηρέαζαν το περιβάλλον πριν την εμφάνιση της ανθρώπινης ζωής, ενώ ενδέχεται να διαδραμάτισαν κρίσιμο ρόλο στην εξέλιξη της ζωής του πλανήτη (Rakov, 2016).

## 1.1. Κεραυνοί

Ο κεραυνός αποτελεί ένα φυσικό φαινόμενο, το οποίο σχετίζεται με την ηλεκτρική δραστηριότητα στην ατμόσφαιρα (Giancoli, 1998). Ως κεραυνός χαρακτηρίζεται μια ισχυρή ηλεκτρική εκκένωση υψηλής εντάσεως, της οποίας το μήκος διαδρομής μετριέται σε χιλιόμετρα (Uman, 1984). Γενικά, μία ηλεκτρική εκκένωση πραγματοποιείται ανάμεσα σε περιοχές με διαφορές φορτίου, η οποία εκδηλώνεται είτε μεταξύ ενός νέφους και του εδάφους είτε μεταξύ δύο τμημάτων του ίδιου νέφους ή μεταξύ δύο διαφορετικών νεφών. Με αυτόν τον τρόπο, εξουδετερώνονται προσωρινά αυτές οι περιοχές μέσω της σχεδόν στιγμιαίας απελευθέρωσης ενέργειας που κυμαίνεται κατά μέσο όρο από εκατοντάδες μεγατζάουλ έως αρκετά γιγατζάουλ, ανάλογα με τον τύπο και την ένταση του φαινομένου.

Αυτή η εκκένωση προκαλεί έντονη λάμψη φωτός και ακολουθείται από τον ήχο της βροντής, ο οποίος προκύπτει από την απότομη διαστολή του θερμαινόμενου αέρα λόγω της εκκένωσης. Σημαντικό είναι ακόμη να αναφερθεί, πως ως αστραπή χαρακτηρίζεται η λάμψη που παρατηρείται κατά τη διάρκεια της ηλεκτρικής εκκένωσης, ενώ ως κεραυνός αναφέρεται η ίδια η ηλεκτρική εκκένωση.

Ο κεραυνός εμφανίζεται συνήθως κατά τη διάρκεια καταιγίδων, ενώ μπορεί να προκληθεί και από ηφαιστειακή δραστηριότητα κατά τη διάρκεια ηφαιστειακών εκρήξεων. Επιπλέον, οι κεραυνοί εκλύουν τεράστιες ποσότητες θερμότητας. Μία μόνο εκκένωση μπορεί να θερμάνει τον αέρα γύρω της σε θερμοκρασίες πέντε φορές υψηλότερες από αυτές της επιφάνειας του ήλιου (Rakov, 2016). Η ταχεία αυτή θέρμανση προκαλεί βίαιη διαστολή

του αέρα, δημιουργώντας το χαρακτηριστικό ήχο της βροντής, μετά την εκδήλωση της λάμψης.



*Σχήμα 1: Κεραυνοί (NASA)*

## **1.2. Δημιουργία κεραυνών**

Ο κεραυνός προκαλείται, όταν η διαφορά τάσης μεταξύ των νεφών και της Γης ή μεταξύ διαφορετικών τμημάτων ενός νέφους γίνεται τόσο υψηλή, ώστε να υπερβεί τη διηλεκτρική αντοχή του αέρα. Το ισχυρό ηλεκτρικό πεδίο επιταχύνει τη κίνηση των ελεύθερων ηλεκτρονίων στον αέρα, δίνοντάς τους αρκετή κινητική ενέργεια ώστε να ιονίσουν τα μόρια του αέρα μέσω συγκρούσεων, αποσπώντας τα ηλεκτρόνια από τα άτομα του αέρα (Giancoli, 1998). Αυτό δημιουργεί μια αλυσιδωτή αντίδραση ιονισμού, σχηματίζοντας μια αγώγιμη διαδρομή πλάσματος. Στη συνέχεια, το ηλεκτρικό ρεύμα ρέει μέσα από αυτήν τη διαδρομή, δημιουργώντας το έντονο φως και τον ήχο που χαρακτηρίζουν τον κεραυνό.

## **1.3. Τύποι κεραυνών**

Όταν η διαφορά δυναμικού, αντίθετα ηλεκτρικά φορτισμένων περιοχών, γίνει

αρκετά μεγάλη, ώστε να υπερνικήσει την αντίσταση του αέρα, δημιουργείται η ηλεκτρική εκκένωση. Αυτή η εκκένωση διακρίνεται ανάλογα με την εμφάνισή της:

1. Μέσα στο ίδιο το νέφος
2. Μεταξύ διαφορετικών νεφών
3. Μεταξύ του νέφους και του εδάφους
4. Μεταξύ του εδάφους και του νέφους (Mohyedin, 2017)

Επιπλέον, οι τύποι των κεραυνών από νέφος προς έδαφος και από έδαφος προς νέφος, μπορούν να ταξινομηθούν με βάση την πολικότητά τους ως εξής: καθοδικός αρνητικός κεραυνός, ανοδικός αρνητικός κεραυνός, καθοδικός θετικός κεραυνός και ανοδικός θετικός κεραυνός (Mohyedin, 2017).

Όσον αφορά την πολικότητα ενός κεραυνού, αυτός δύναται να είναι είτε θετικός είτε αρνητικός. Θετικός θεωρείται ο κεραυνός, ο οποίος μεταφέρει θετικά φορτία, ενώ είναι σπανιότερος και πιο ισχυρός από τον αρνητικό, ο οποίος μεταφέρει αρνητικά φορτία στη γη και είναι ο πιο συνήθης.

Ένας κεραυνός διαχωρίζεται ακόμη ανάλογα με τη κατεύθυνση του. Χαρακτηρίζεται ως «κατερχόμενος» όταν εξελίσσεται από τα ανώτερα στρώματα του νέφους προς το έδαφος, ενώ ακολουθώντας την αντίθετη πορεία, ονομάζεται «ανερχόμενος».

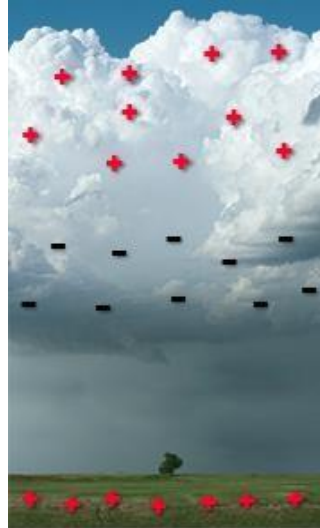
#### *1.3.1. Μέσα στο ίδιο το νέφος*

Μία ηλεκτρική εκκένωση, στις περισσότερες περιπτώσεις, εκδηλώνεται εντός των καταιγιδοφόρων νεφών (Uman, 2001), στα οποία οι συγκρούσεις μεταξύ σωματιδίων βροχής, πάγου και χιονιού δημιουργούν συσσώρευση ηλεκτρικών φορτίων.

Κατά τη διάρκεια μιας καταιγίδας παρατηρείται έντονη ηλεκτρική ανισορροπία ανάμεσα στα στρώματα ενός καταιγιδοφόρου νέφους, που συνοδεύεται από ισχυρές αναταράξεις, με ανοδικά και καθοδικά ρεύματα σε μικρή απόσταση μεταξύ τους. Τα ανοδικά ρεύματα μεταφέρουν τις σταγόνες νερού από τα χαμηλότερα στρώματα της ατμόσφαιρας, ενώ τα καθοδικά ρεύματα μεταφέρουν χαλάζι και πάγο από τα ανώτερα στρώματα προς τα χαμηλότερα. Όπως αναφέρθηκε, εντοπίζονται συγκρούσεις μεταξύ σταγονιδίων νερού, πάγου και χαλαζιού μέσα στο νέφος. Κατά τη διάρκεια αυτών των



συγκρούσεων, τα ανερχόμενα σωματίδια χάνουν ηλεκτρόνια, τα οποία συγκεντρώνονται στα κατερχόμενα σωματίδια και δημιουργούν συσσώρευση ηλεκτρικών φορτίων.



*Σχήμα 2: Τα φορτία ενός νέφους κατά τη διάρκεια καταιγίδας (NOAA)*

Ως αποτέλεσμα, το σύννεφο αποκτά θετικά φορτία στην κορυφή και αρνητικά στη βάση του. Επομένως, αυτές οι συγκρούσεις προκαλούν διαχωρισμό των ηλεκτρικών φορτίων και την εμφάνιση ηλεκτρικού πεδίου μέσα στη καταιγίδα, δημιουργώντας έτσι, μια διαφορά δυναμικού μεταξύ διαφορετικών περιοχών του σύννεφου. Όταν η διαφορά αυτή γίνει αρκετά μεγάλη, προκαλείται μια ηλεκτρική εκκένωση, δηλαδή το φαινόμενο του κεραυνού.

#### *1.3.2. Μεταξύ διαφορετικών νεφών*

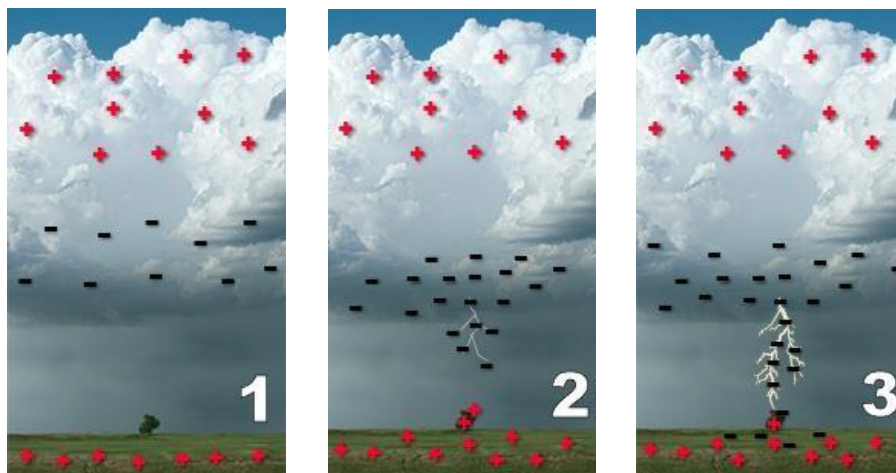
Με την ύπαρξη δύο κοντινών καταιγιδοφόρων νεφών, είναι δυνατόν να σχηματιστεί ηλεκτρική εκκένωση μεταξύ τους. Αυτό συμβαίνει συνήθως σε συμπλέγματα καταιγίδων και είναι πιο σπάνιο φαινόμενο σε σχέση με τους άλλους τύπους.

#### *1.3.3. Μεταξύ του νέφους και του εδάφους*

Αντίστοιχα, παρόμοια διαδικασία συμβαίνει ανάμεσα στο νέφος και το έδαφος. Οι χαμηλότερες περιοχές των νεφών φορτίζονται αρνητικά, ενώ αντικείμενα στο έδαφος φορτίζονται θετικά. Αυτή η ανισορροπία του ηλεκτρικού φορτίου έχει ως αποτέλεσμα τη δημιουργία κεραυνών, καθώς η φύση προσπαθεί να εξισορροπήσει αυτά τα φορτία.

Ο κεραυνός είναι δυνατόν να σχηματιστεί μέσω τεσσάρων διαδικασιών, οι οποίες είναι ο διαχωρισμός φορτίων, ο σχηματισμός ηγέτη, η εκκένωση και η επανεκκίνηση (Mohyedin, 2017).

Πιο συγκεκριμένα, ο σχηματισμός του «ηγέτη» περιλαμβάνει τη δημιουργία μίας διαδρομής που αποτελείται από αρνητικά φορτία και κατέρχεται από το νέφος προς το έδαφος (Uman, 2001). Ένας αρνητικά φορτισμένος ηγέτης ονομάζεται «κλιμακωτός ηγέτης», ενώ μεταδίδεται με μικρά βήματα, σχηματίζοντας διακλαδώσεις. Όταν οι κλιμακωτοί ηγέτες πλησιάζουν το έδαφος, η παρουσία αντίθετων φορτίων στο έδαφος ενισχύει το ηλεκτρικό πεδίο. Ένα αρκετά ισχυρό ηλεκτρικό πεδίο μπορεί να αναπτύξει μια αγωγίμη εκκένωση. Όταν οι δύο ηγέτες συναντηθούν, το ηλεκτρικό ρεύμα αυξάνεται σημαντικά. Όταν ο ηγέτης πλησιάσει αρκετά σε ένα θετικά φορτισμένο αντικείμενο στο έδαφος, δημιουργείται ένας «ακροδέκτης», ο οποίος ανεβαίνει από το έδαφος προς το νέφος, ώστε να συναντήσει των κατερχόμενο αγωγό. Μόλις συνδεθούν τα δύο κανάλια, δημιουργείται η επιστροφή του ρεύματος, μια έντονη λάμψη που κινείται προς τα πάνω με τεράστια ταχύτητα και ολοκληρώνει την ηλεκτρική εκκένωση. Η διαδικασία, στην οποία οι κεραυνοί αποτελούνται από πολλαπλά χτυπήματα ονομάζεται επανεκκίνηση. Μία τυπική εκκένωση ενός κεραυνού αποτελείται από τρία έως τέσσερα χτυπήματα, αλλά μπορεί να είναι και περισσότερα. Οι διαδοχικές εκκενώσεις ενός κεραυνού προκαλούν τον ήχο της βροντής (Mohyedin, 2017).



Σχήμα 3: Τα στάδια δημιουργίας ενός κεραυνού μεταξύ νέφους και εδάφους (NOAA)

Οι κεραυνοί που καταλήγουν στο έδαφος διακρίνονται σε αρνητικούς ή θετικούς, ανάλογα με την κατεύθυνση του ηλεκτρικού ρεύματος. Οι κεραυνοί στις πλειοψηφία τους, είναι αρνητικοί, στους οποίους ένα αρνητικό φορτίο μεταφέρεται στο έδαφος και τα ηλεκτρόνια κινούνται προς τα κάτω κατά μήκος του καναλιού του κεραυνού. Στους θετικούς κεραυνούς, τα ηλεκτρόνια κινούνται προς τα πάνω κατά μήκος του καναλιού και ένα θετικό φορτίο μεταφέρεται στο έδαφος (Sargoytchev, 2019).

#### **1.4. Το φαινόμενο της βροντής**

Η έντονη θέρμανση του αέρα γύρω από την πορεία του κεραυνού προκαλεί τη ραγδαία διαστολή του, δημιουργώντας ένα ηχητικό κύμα που ακούγεται ως βροντή. Συνήθως, ο ήχος αυτός καθυστερεί, σε σχέση με τη λάμψη, λόγω της διαφορετικής ταχύτητας του φωτός και του ήχου.

#### **1.5. Σημείο κρούσης κεραυνού**

Αντικείμενα που βρίσκονται πιο κοντά στη βάση του σύννεφου, τα ψηλότερα στην ευρύτερη περιοχή, είναι περισσότερο ευάλωτα και συνήθως στόχοι των κεραυνών. Αυτό είναι αποτέλεσμα της σημαντικής διαφοράς δυναμικού που δημιουργείται μεταξύ των φορτισμένων σύννεφων και αυτών των αντικειμένων, τα οποία λειτουργούν ως αγωγοί για αυτή την εκφόρτιση. Ωστόσο, η ακριβής περιοχή που θα χτυπήσει ένας κεραυνός εξαρτάται από τη συγκέντρωση των ηλεκτρικών φορτίων και επομένως οι κεραυνοί δεν περιορίζονται στα υψηλότερα αντικείμενα αλλά μπορεί να χτυπήσουν απευθείας στο έδαφος σε ανοιχτούς χώρους, με τη προϋπόθεση πως σε εκείνη τη περιοχή συγκεντρώνεται επαρκές θετικό φορτίο για την εκφόρτιση. Έτσι, κρίνεται απαραίτητο να αποφεύγεται η παραμονή κοντά σε υψηλά αντικείμενα, καθώς και σε εκτεθειμένες ανοιχτές περιοχές κατά τη διάρκεια μιας καταιγίδας.

## 1.6. Κίνδυνοι και Επιπτώσεις

Ο κεραυνός μπορεί να αποτελέσει ένα πολύ επικίνδυνο φαινόμενο με σημαντικές επιπτώσεις. Όσον αφορά την ανθρώπινη ζωή, καταγράφονται περίπου 2.000 θάνατοι ετησίως, ενώ οι επιζώντες είναι πολύ πιθανό να αντιμετωπίσουν μακροχρόνιες βλάβες, όπως νευρολογικά προβλήματα. Εκτός από την ανθρώπινη ζωή, αναφέρονται τραυματισμοί και θάνατοι και σε ζώα. Προκύπτουν σημαντικές υλικές ζημιές αλλά και καταστροφές, καθώς οι κεραυνοί είναι δυνατόν να προκαλέσουν πυρκαγιές, βλάβες στην ανθρώπινη περιουσία και σε ηλεκτρικά δίκτυα. Εντυπωσιακό είναι το γεγονός πως η θερμότητα που εκλύεται από τους κεραυνούς, είναι ικανή να εξατμίσει το νερό που βρίσκεται μέσα στους κορμούς των δέντρων, δημιουργώντας τεράστια πίεση και προκαλώντας την έκρηξή τους και συνεπώς δημιουργώντας τις κατάλληλες συνθήκες για μία δασική πυρκαγιά.

Οι κεραυνοί, όπως αναφέρθηκε, αποτελούν σημαντικό παράγοντα πρόκλησης πυρκαγιών, με τις δασικές πυρκαγιές να συγκαταλέγονται στις σοβαρότερες και πιο επικίνδυνες, οδηγώντας σε σημαντικές εκπομπές ρύπων, όπως μαύρο άνθρακα και μεθάνιο, που αποτελούν βραχύβιους κλιματικούς παράγοντες (Whaley et al., 2024). Εκτός από την οικολογική καταστροφή που προκαλούν, οι κεραυνοί ευθύνονται για πάνω από το 30% των βλαβών σε ηλεκτρικές γραμμές υψηλής τάσεως, επηρεάζοντας κρίσιμες υποδομές, ενώ κάθε εμπορικό αεροσκάφος πλήττεται από κεραυνό κατά μέσο όρο μία φορά ετησίως (Rakov, 2016).



Σχήμα 4: Κρούση κεραυνού σε κορμό δέντρου (Monster Tree Service)

Ωστόσο, ένας κεραυνός είναι δυνατόν να βελτιώσει την ποιότητα του αέρα, αυξάνοντας την ικανότητα της ατμόσφαιρας να καθαρίζεται, παράγοντας μονοξείδιο του αζώτου (NO), το οποίο οδηγεί σε χημικές διεργασίες που σχηματίζουν όζον (O<sub>3</sub>) και τον κύριο οξειδωτικό παράγοντα της ατμόσφαιρας, τη ρίζα υδροξυλίου (OH) (Brune et al., 2021).

## 1.7. Εφαρμογές αναγνώρισης κεραυνών

Η αναγνώριση και παρακολούθηση κεραυνών βρίσκει εφαρμογή σε διάφορους τομείς, παρέχοντας σημαντικά δεδομένα για την πρόληψη κινδύνων αλλά και την υποστήριξη της επιστημονικής έρευνας.

Στον τομέα της μετεωρολογίας, η έγκαιρη αναγνώριση και παρακολούθηση καταιγίδων συμβάλλει στην πρόβλεψη της εμφάνισης κεραυνών. Ωστόσο, είναι αδύνατο να προβλεφθούν μεμονωμένοι κεραυνοί, καθώς ο κεραυνός χαρακτηρίζεται από τη τυχαία εμφάνισή του, την εξάπλωση και τη συχνότητα (NOAA National Severe Storms Laboratory). Με αυτόν τον τρόπο, παρέχονται σημαντικές προειδοποιήσεις, οι οποίες μειώνουν την έκθεση της ζωής και της περιουσίας σε κίνδυνο.



Σχήμα 5: Κρούση κεραυνού σε δέντρο (panoramatreeservice)

Όπως αναφέρθηκε, οι κεραυνοί αποτελούν αιτία πρόκλησης δασικών πυρκαγιών, καθώς τα περιβάλλοντα αυτά θεωρούνται ευνοϊκά για την έναρξη και εξάπλωση πυρκαγιών. Ένα σύστημα ανίχνευσης κεραυνών συμβάλλει στην άμεση ενημέρωση και έγκαιρη ανίχνευση πυρκαγιών σε αυτά τα περιβάλλοντα, ενώ εξαλείφει την ανάγκη για αρχικές έρευνες περιοχών με αεροσκάφη ή επίγειες ομάδες, εντοπίζοντας έγκαιρα τα σημεία που έχουν πληγεί από κεραυνούς (Holle, 1993).

Στην αεροπορία, η ανάλυση δεδομένων για κεραυνούς είναι κρίσιμη για την ασφάλεια των πτήσεων (Fisher and Plummer, 1980). Ο εντοπισμός των περιοχών με έντονη ηλεκτρική δραστηριότητα είναι απαραίτητη για τον σχεδιασμό ασφαλέστερων διαδρομών για τα αεροσκάφη, μειώνοντας τον κίνδυνο πρόσκρουσης από κεραυνούς και διασφαλίζοντας την ασφάλεια των επιβατών και των πληρωμάτων.

Στην βιομηχανία εξόρυξης, είναι απαραίτητη η ανίχνευση και άμεση ενημέρωση σχετικά με την εμφάνιση κεραυνών, καθώς ένας κεραυνός μπορεί να προκαλέσει εκρήξεις στα καύσιμα ή άλλα εύφλεκτα υλικά στο όχημα, θέτοντας σε κίνδυνο την ασφάλεια των εργαζομένων και των εγκαταστάσεων (Low and Zhou, 2022). Ακόμη, ένας κεραυνός μπορεί να προκαλέσει έκρηξη μεθανίου μέσα σε ένα υπόγειο μεταλλευτικό ορυχείο, για αυτό η ανίχνευσή του είναι σημαντική για τη διασφάλιση της ασφάλειας των μεταλλωρύχων και την πρόληψη σοβαρών ατυχημάτων (Kithil, 2014).

Όσον αφορά την προστασία κρίσιμων υποδομών, ο εντοπισμός κεραυνών είναι απαραίτητος για την προστασία ευαίσθητων εγκαταστάσεων, όπως ηλεκτρικά δίκτυα αλλά και κτηρίων και των ηλεκτρικών συσκευών που βρίσκονται μέσα σε αυτά (Bullis et al., 2023). Η έγκαιρη αναγνώριση κεραυνών επιτρέπει τη λήψη μέτρων, τα οποία είναι δυνατόν να αποτρέψουν τυχόν καταστροφικές συνέπειες.

Η μελέτη των κεραυνών αποτελεί σημαντικό εργαλείο για την κατανόηση του κλίματος και της κλιματικής αλλαγής (Price, 2009). Τα δεδομένα που συλλέγονται γίνονται αντικείμενο μελέτης επιστημονικών ερευνών, ενισχύοντας τη γνώση για τα φυσικά φαινόμενα και το περιβάλλον.

Επιπλέον, η αναγνώριση κεραυνών έχει εφαρμογές και σε τομείς όπως η ναυτιλία, στην οποία συμβάλλει στην ασφαλή πλοήγηση πλοίων σε περιοχές με έντονη ηλεκτρική δραστηριότητα (Raunek, 2021). Παράλληλα, χρησιμοποιείται και σε περιβαλλοντικές μελέτες για την παρακολούθηση της ηλεκτρικής δραστηριότητας σε ευαίσθητα οικοσυστήματα, όπως είναι τα τροπικά δάση (Yanoviak et al., 2024).

Συνολικά, η αξιοποίηση της τεχνολογίας αναγνώρισης κεραυνών αποτελεί ένα ισχυρό εργαλείο για την πρόληψη κινδύνων, την ενίσχυση της ασφάλειας και την προώθηση της επιστημονικής κατανόησης του φυσικού περιβάλλοντος.

## **1.8. Προστασία από κεραυνούς**

Από τα πιο διαδεδομένα μέσα προστασίας έναντι των κεραυνών είναι η χρήση αλεξικέραυνων και αγωγών στα κτίρια, τα οποία στοχεύουν στη διοχέτευση της ηλεκτρικής εκκένωσης με ασφαλή τρόπο στο έδαφος. Βέβαια, το νερό που βρίσκεται στους αγωγούς αλλά και οι σταθερές τηλεφωνικές γραμμές μπορούν να μεταφέρουν ρεύμα. Τα αυτοκίνητα προσφέρουν ασφάλεια λόγω της μεταλλικής δομής τους που κατευθύνει το ρεύμα προς το έδαφος.

Ο κεραυνός είναι ένα εντυπωσιακό αλλά επικίνδυνο φυσικό φαινόμενο, ενώ ένα χτύπημά του είναι ικανό να προκαλέσει σοβαρές συνέπειες. Για αυτό το λόγο απαιτείται προσοχή και κατάλληλα μέτρα προστασίας για την αποφυγή δυσάρεστων επιπτώσεων, τόσο σε επίπεδο υποδομών όσο και σε επίπεδο ατομικής ασφάλειας.

## **1.9. Μέθοδοι ανίχνευσης κεραυνών**

Η ανίχνευση κεραυνών αναφέρεται στη διαδικασία, η οποία καταγράφει την ηλεκτρική εκκένωση και βασίζεται στην παρατήρηση των φυσικών φαινομένων που συνοδεύουν τους κεραυνούς, μέσω διαφόρων συστημάτων. Τέτοια φαινόμενα μπορεί να είναι ηλεκτρομαγνητικά και ακουστικά κύματα αλλά και οπτικά σήματα. Μία συσκευή ανίχνευσης κεραυνών είναι ικανή να ανιχνεύει μία ηλεκτρική εκκένωση, με τις συσκευές αυτές να αποτελούν συστήματα εδάφους, κινητά και δορυφορικά συστήματα, τα οποία εντοπίζουν την κατεύθυνση της εκκένωσης από την τρέχουσα τοποθεσία τους.

Τα συστήματα εδάφους χρησιμοποιούν τριγωνισμό από πολλαπλές τοποθεσίες για να καθορίσουν την απόσταση (NASA).

Η ανίχνευση της ηλεκτρικής εκκένωσης με δορυφορικά μέσα χρησιμοποιείται για τον εντοπισμό της κατεύθυνσης, της απόστασης και της έντασης του φαινομένου αυτού μέσω άμεσης παρατήρησης (NASA).

#### *1.9.1. Ανίχνευση μέσω ηλεκτρομαγνητικών κυμάτων*

Η ανίχνευση μιας εκκένωσης μέσω ηλεκτρομαγνητικών κυμάτων επιτυγχάνεται με τη χρήση ηλεκτρομαγνητικών αισθητήρων. Οι αισθητήρες αυτοί ανιχνεύουν το ηλεκτρομαγνητικό σήμα που εκπέμπεται από την εκκένωση του κεραυνού κατά τη διάρκεια μίας ηλεκτρικής εκκένωσης, όταν αυτή έρχεται σε επαφή με το έδαφος. Αυτή η μέθοδος ανίχνευσης είναι ευρέως διαδεδομένη και χρησιμοποιείται τόσο από δημόσιους όσο και από ιδιωτικούς φορείς.

##### *1.9.1.1. Το δίκτυο EUCLID*

Η EUCLID αποτελεί μια συνεργασία εθνικών δικτύων ανίχνευσης κεραυνών με σκοπό την αναγνώριση και ανίχνευση κεραυνών σε όλη την Ευρώπη (EUCLID). Το δίκτυο περιλαμβάνει περίπου 164 ηλεκτρομαγνητικούς αισθητήρες εγκατεστημένους σε όλες τις ευρωπαϊκές χώρες, με την κάλυψή του να αποτυπώνεται στην παρακάτω εικόνα (Σχήμα 6).



*Σχήμα 6: Χάρτης κάλυψης περιοχών του δικτύου EUCLID (EUCLID)*



Οι αισθητήρες του δικτύου αυτού, καταγράφουν την ηλεκτρομαγνητική ακτινοβολία που εκπέμπεται με την εκκένωση του κεραυνού και στη συνέχεια, γίνεται η επεξεργασία των δεδομένων αυτών. Για κάθε κεραυνό καταγράφονται πληροφορίες, όπως η χρονική στιγμή του συμβάντος, η γεωγραφική θέση, η ένταση και η πολικότητα του ρεύματος, καθώς και ο αριθμός των επαναλαμβανόμενων εκκενώσεων. Η ανίχνευση κεραυνών γίνεται με τη χρήση της συνδυαστικής μεθόδου, με τη χρήση δορυφορικών σημάτων GPS (EUCLID).

#### 1.9.1.2. Το σύστημα ZEYΣ

Το σύστημα ZEYΣ αποτελεί ένα δίκτυο ανίχνευσης ηλεκτρικών εκκενώσεων του Εθνικού Αστεροσκοπείου Αθηνών. Το δίκτυο περιλαμβάνει έξι δέκτες στην Ευρώπη, οι οποίοι παρατηρούνται στη παρακάτω εικόνα (Σχήμα 7) (Εθνικό Αστεροσκοπείο Αθηνών).



Σχήμα 7: Το σύστημα ZEYΣ (METEO)

Το σύστημα ZEYΣ, σύμφωνα με το Εθνικό Αστεροσκοπείο Αθηνών αξιοποιεί την ιδιότητα ότι κάθε ηλεκτρική εκκένωση παράγει μια μοναδική κυματομορφή, γεγονός που επιτρέπει τη διάκριση μεταξύ διαφορετικών γεγονότων. Ωστόσο, για τον ακριβή εντοπισμό της θέσης μιας εκκένωσης μεταξύ νέφους και εδάφους, απαιτούνται τέσσερις σταθμοί.

Όταν ένας κεραυνός χτυπά το έδαφος, η κυματομορφή που εκπέμπεται καταγράφεται από όλους τους επίγειους σταθμούς σε διαφορετικούς χρόνους, ανάλογα με την απόστασή τους από το σημείο της εκκένωσης. Το κέντρο ελέγχου του συστήματος ορίζει έναν σταθμό ως «σταθμό αναφοράς» και του αποδίδει μηδενική διαφορά χρόνου άφιξης του σήματος. Στη συνέχεια, υπολογίζονται οι διαφορές χρόνου άφιξης για τους υπόλοιπους σταθμούς, συγκριτικά με τον σταθμό αναφοράς.

Η ανάλυση αυτή χρησιμοποιείται για να εντοπιστεί ο κοινός γεωμετρικός τόπος των σημείων, όπου η διαφορά άφιξης του σήματος μεταξύ του σταθμού αναφοράς και ενός άλλου σταθμού είναι η ίδια. Τα σημεία αυτά σχηματίζουν μια υπερβολή. Η διαδικασία επαναλαμβάνεται για όλους τους σταθμούς του δικτύου και η τοποθεσία της ηλεκτρικής εκκένωσης προσδιορίζεται ως το σημείο τομής όλων των υπερβολών.

#### *1.9.1.3. Ακουστικά κύματα*

Συστήματα βασισμένα στην ανίχνευση ακουστικών σημάτων έχουν την ικανότητα να ανιχνεύουν τα ηχητικά κύματα των ηλεκτρικών εκκενώσεων, αλλά χρησιμοποιούνται συνήθως συμπληρωματικά, ώστε να παρέχουν πλήρη δεδομένα σχετικά με τη δραστηριότητα αυτών. Τέτοια συστήματα χρησιμοποιούνται μεμονωμένα σε εξειδικευμένες εφαρμογές. Μία τέτοια εφαρμογή, αποτελεί η ερευνητική μελέτη με σκοπό την ανάπτυξη ενός οικονομικού και αξιόπιστου συστήματος ανίχνευσης μίας ηλεκτρικής εκκένωσης για την προστασία κρίσιμων υποδομών, όπως οι ανεμογεννήτριες στην Ιαπωνία.

Καθώς οι ανεμογεννήτριες είναι ευάλωτες στους κεραυνούς, στις παράκτιες περιοχές της Ιαπωνίας έχουν εγκατασταθεί συστήματα ανίχνευσης κεραυνών που διακόπτουν αμέσως τη λειτουργία των ανεμογεννητριών όταν εντοπιστεί εκκένωση σε αυτήν, για την αποτροπή ζημιών και την εξασφάλιση της ασφάλειας.

Η μέθοδος ανίχνευσης της μελέτης αυτής, χρησιμοποιεί ακουστικά δεδομένα από κεραυνούς για να καθορίσει αν έχει σημειωθεί εκκένωση και προτείνεται ως μια πιο οικονομική και ακριβής εναλλακτική σε σχέση με τα υπάρχοντα συστήματα ανίχνευσης, τα οποία συνήθως είναι δαπανηρά και βασίζονται κυρίως σε ηλεκτρομαγνητική ανίχνευση (Higashi and Yamamoto, 2023).

#### *1.9.1.4. Οπτικά σήματα*

Η ανίχνευση κεραυνών μέσω οπτικών σημάτων χρησιμοποιεί οπτικούς αισθητήρες για την παρακολούθηση και καταγραφή της εκκένωσης. Παράδειγμα αυτής της μεθόδου ανίχνευσης αποτελεί ο Γεωστατικός Χαρτογράφος Κεραυνών (GLM). Ο GLM χαρτογραφεί

τη συνολική δραστηριότητα των εκκενώσεων τόσο εντός νέφους όσο και από το νέφος προς το έδαφος, πάνω από την Αμερική και τις παρακείμενες θαλάσσιες περιοχές. Αποτελεί ένα όργανο που βρίσκεται σε δύο δορυφόρους με σκοπό την παρατήρηση του οπτικού σήματος, την φωτεινή εκπομπή, που συνοδεύει την εκκένωση. Βέβαια, παρουσιάζει ένα μειονέκτημα καθώς δεν είναι δυνατόν να καταγραφεί εάν το κανάλι της εκκένωσης φτάνει στο έδαφος ή όχι (NOAA NSSL).

Η ανίχνευση κεραυνών μέσω οπτικών σημάτων προσφέρει αρκετά πλεονεκτήματα, όπως η δυνατότητα κάλυψης απομακρυσμένων και δυσπρόσιτων περιοχών, στις οποίες η εγκατάσταση επίγειων αισθητήρων είναι δύσκολη. Ωστόσο, η αποτελεσματικότητα αυτής της μεθόδου μπορεί να επηρεαστεί από διάφορους παράγοντες λόγω των ατμοσφαιρικών συνθηκών (NOAA).

## 2. Παραδοσιακές μέθοδοι αναγνώρισης κεραυνών

Οι παραδοσιακές μέθοδοι αναγνώρισης κεραυνών βασίζονται στις αρχές της φυσικής, των μαθηματικών αλλά και στις κλασικές τεχνικές επεξεργασίας σήματος και εικόνας.

### 2.1. Μέθοδοι με fractals

Ο όρος φράκταλ (fractal), σύμφωνα με τον Mandelbrot το 1975, αναφέρεται σε μία γεωμετρία «σπασμένη» ή «κατακερματισμένη», ενώ επεκτείνοντας αυτή τη γεωμετρία σε σχέση με τους κεραυνούς, δηλώνει πως η διαδρομή του κεραυνού δεν είναι μια απλή ευθεία γραμμή, αλλά μια χαοτική πορεία με πολλές διακλαδώσεις. Οι κεραυνοί κινούνται με τρόπο που συσχετίζεται με τη δομή φράκταλ, καθώς η ηλεκτρική τους εκκένωση εξαπλώνεται σε πολύπλοκες διαδρομές μέσα στην ατμόσφαιρα.

Καθώς τα φράκταλ είναι περιπλοκότητα στη μορφή τους, δεν μπορούν να μελετηθούν σε σχέση με τα συμβατικά γεωμετρικά σχήματα. Κατά συνέπεια, δεν είναι δυνατόν να οριστούν με τη βοήθεια της ευκλείδειας γεωμετρίας (Britannica, 2025).

Ένα χαρακτηριστικό παράδειγμα της fractal δομής, το οποίο παρατηρείται στην φύση, είναι ο κεραυνός. Όπως ανέφερε ο Benoit Mandelbrot, οι κεραυνοί δεν ταξιδεύουν σε ευθείες γραμμές, αλλά ακολουθούν χαοτικές, τεθλασμένες διαδρομές, οι οποίες δημιουργούνται από τη διάσπαση του ηλεκτρικού φορτίου που έχει συσσωρευτεί στην ατμόσφαιρα.

Μία προσέγγιση για τη μαθηματική ανάλυση του κεραυνού αποτελεί η προσέγγισή του ως φράκταλ δομής με απόσβεση περιοδικής ταλάντωσης, με την εισαγωγή της συχνότητας ταλάντωσης και του χρόνου χαλάρωσης. Η ακτινοβολία που εκπέμπεται από έναν κεραυνό μπορεί να φτάσει σε πολύ μεγαλύτερες αποστάσεις από το ίδιο το ορατό του μήκος, γεγονός που εξηγείται μέσω της φρακταλικής του φύσης. Ωστόσο, η στοχαστική φύση της διάδοσης των εκκενώσεων και η πολυπλοκότητα της γεωμετρίας του κεραυνού δυσκολεύουν την ακριβή πρόβλεψη της συμπεριφοράς του. Παρά το γεγονός ότι η συνολική απόσταση που διανύει μπορεί να είναι πολύ μεγάλη, η πραγματική απόσταση σε ευθεία μεταξύ του σημείου εκκίνησης και του προορισμού είναι πολύ μικρότερη (Balkhanov and Bashkuev, 2012).

Η στοχαστική φύση της διάδοσης των εκκενώσεων, δημιουργεί πολλαπλές διακλαδώσεις και καμπυλώσεις, με αποτέλεσμα να καθιστά πολλά συστήματα προστασίας αναποτελεσματικά, καθώς πλευρικές εκκενώσεις μπορεί να προκαλέσουν ζημιές σε κοντινές κατασκευές. Επίσης η ακριβής τιμή της φρακταλικής διάστασης των κεραυνών παρουσιάζει σημαντικές διακυμάνσεις στις προσομοιώσεις, λόγω της έλλειψης καθορισμένων θεωρητικών μοντέλων για την επιλογή των κατάλληλων παραμέτρων (Nguyen et al., 2001).

Από το 2001 έως σήμερα, η μελέτη των φρακταλικών ιδιοτήτων των κεραυνών έχει επεκταθεί σημαντικά, μέσω της ανάπτυξης προηγμένων προσομοιώσεων και της χρήσης σύγχρονων τεχνικών απεικόνισης. Ωστόσο, η φρακταλική διάσταση των κεραυνών εξακολουθεί να παρουσιάζει διακυμάνσεις στις προσομοιώσεις, καθώς επηρεάζεται από διάφορους παράγοντες, όπως οι περιβαλλοντικές συνθήκες και οι ιδιότητες του μέσου διάδοσης. Παρά την πρόοδο στην κατανόηση του φαινομένου, δεν υπάρχει ακόμη ένα καθολικά αποδεκτό θεωρητικό μοντέλο που να καθορίζει με ακρίβεια τις παραμέτρους για τον υπολογισμό της φρακταλικής διάστασης των κεραυνών.

## **2.2. Ντετερμινιστικοί μαθηματικοί αλγόριθμοι**

Οι ντετερμινιστικοί αλγόριθμοι βασίζονται σε αυστηρούς μαθηματικούς κανόνες και δεν εμπεριέχουν στοχαστικά στοιχεία. Χρησιμοποιούνται συχνά στη φυσική και τη μετεωρολογία για την ανάλυση φαινομένων, όπως είναι και οι κεραυνοί. Βασίζονται σε μαθηματικά μοντέλα και αριθμητικές μεθόδους για την επίλυση εξισώσεων που περιγράφουν τη δυναμική του ηλεκτρικού πεδίου και των εκκενώσεων στην ατμόσφαιρα.

## **2.3. Μέθοδοι βασισμένες σε διαφορικές εξισώσεις**

Οι μέθοδοι βασισμένες σε διαφορικές εξισώσεις χρησιμοποιούνται για την περιγραφή των φυσικών διεργασιών που διέπουν τους κεραυνούς. Παράδειγμα αυτής της μεθόδου, αποτελούν οι εξισώσεις του Maxwell στη μελέτη ηλεκτρικών εκκενώσεων (Thiemann and Gasiewski, 2014). Οι κεραυνοί αποτελούν ηλεκτρικά φαινόμενα που περιγράφονται από αυτές τις εξισώσεις, οι οποίες καθορίζουν τη συμπεριφορά των

ηλεκτρομαγνητικών πεδίων (Meredith and Earles, 2010), ενώ βρίσκουν εφαρμογή σε μοντέλα ανάπτυξης ηλεκτρικών εκκενώσεων.

#### *2.3.1. Αριθμητική ανάλυση και πεπερασμένα στοιχεία*

Οι διαφορικές εξισώσεις που περιγράφουν τους κεραυνούς συνήθως δεν παρέχουν αναλυτικές λύσεις, οπότε γίνεται χρήση αριθμητικών μεθόδων, όπως η μέθοδος πεπερασμένων διαφορών στο πεδίο του χρόνου (Finite Difference Time Domain - FDTD).

Η μέθοδος πεπερασμένων διαφορών στο πεδίο του χρόνου είναι η πιο διαδεδομένη τεχνική για προσομοιώσεις κεραυνών. Επιτρέπει την ακριβή ανάλυση των ηλεκτρομαγνητικών διεργασιών που προκαλεί ένας κεραυνός, όπως, για παράδειγμα, στην περίπτωση ενός αεροσκάφους. Η FDTD αποτελεί βασικό εργαλείο για την προσομοίωση και κατανόηση των ηλεκτρομαγνητικών επιπτώσεων (Stracqualursi et al., 2023).

### **2.4. Αλγόριθμοι και υπολογιστικές μέθοδοι**

Πριν από την ανάπτυξη των προσεγγίσεων βασισμένων σε TN, οι υπολογιστικές μέθοδοι ακολουθούσαν κλασικές αλγοριθμικές αρχές, όπως αναλύονται παρακάτω.

#### *2.4.1. Fast Fourier Transform (FFT)*

Η μέθοδος αυτή χρησιμοποιείται για την ανάλυση των ραδιοσημάτων που παράγονται από κεραυνούς. Στο πλαίσιο του ερευνητικού έργου του Πανεπιστημίου του Νέου Μεξικού (UNM), αναπτύσσεται μια καινοτόμος μέθοδος ανίχνευσης κεραυνών, η οποία βασίζεται σε έναν συνδυασμό δορυφορικών και επίγειων συστημάτων, με στόχο την ακριβή καταγραφή των ηλεκτρομαγνητικών σημάτων που παράγονται από τους κεραυνούς και τη μελέτη της αλληλεπίδρασής τους με την ιονόσφαιρα. Το σύστημα λειτουργεί υπολογίζοντας την παραμόρφωση των παλμών των κεραυνών καθώς αυτοί εκτρέπονται λόγω ιονοσφαιρικής διάθλασης. Για τη διαδικασία ανίχνευσης και ανάλυσης των σημάτων των κεραυνών, αρχικά γίνεται η λήψη και η καταγραφή του σήματος. Τα καταγεγραμμένα δεδομένα αποθηκεύονται και μεταφέρονται σε υπολογιστή για επεξεργασία. Αναλύονται στο MATLAB και χρησιμοποιείται ο μετασχηματισμός Fourier (Fast Fourier Transform

- FFT) για τη μετατροπή των δεδομένων από το πεδίο του χρόνου στο πεδίο της συχνότητας (Ierides et al., 2011).

#### 2.4.2. *Monte Carlo*

Οι Μέθοδοι Monte Carlo χρησιμοποιούνται στη μοντελοποίηση και προσομοίωση του φαινομένου του κεραυνού, επιτρέποντας την ανάλυση της διάδοσης του φωτός της εκκένωσης μέσω της ατμόσφαιρας και των καταιγιδοφόρων νεφών. Για παράδειγμα, το τρισδιάστατο πολωμένο μοντέλο ατμοσφαιρικής μεταφοράς ακτινοβολίας Monte Carlo (3DMCPOL) αποτελεί έναν αλγόριθμο προσομοίωσης βασισμένο στη μέθοδο Monte Carlo και εφαρμόζεται για την προσομοίωση της διάδοσης του φωτός της αστραπής μέσω ετερογενών νεφών σε τρεις διαστάσεις (Rimboud et al., 2024).

### 3. Η χρήση της Τεχνητής Νοημοσύνης στην ανίχνευση κεραυνών

#### 3.1. Επεξεργασία εικόνας

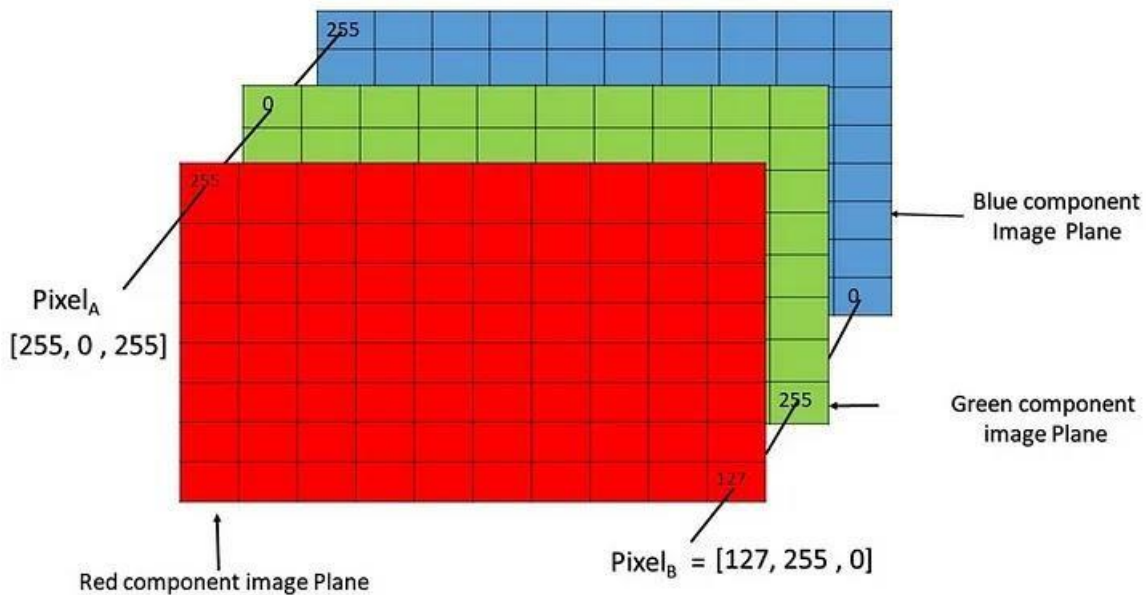
Η εικόνα είναι ένα μέσο, το οποίο περιέχει πλήθος πληροφοριών και αποτελεί μια σημαντική πηγή για την απόκτηση και ανταλλαγή πληροφοριών για τον άνθρωπο (Zhang and Dahu, 2019). Ωστόσο, για την απόκτηση αξιόπιστων πληροφοριών από μια εικόνα, είναι απαραίτητη η επεξεργασία της.

Μία ψηφιακή εικόνα ορίζεται ως μια μαθηματική αναπαράσταση μιας εικόνας σε ψηφιακή μορφή, δηλαδή μέσω αριθμών. Συγκεκριμένα, η ψηφιακή εικόνα αποτελεί μια δισδιάστατη συνάρτηση  $f(x,y)$ , όπου οι  $x$  και  $y$  είναι οι συντεταγμένες της εικόνας στον οριζόντιο και κατακόρυφο άξονα και η τιμή της συνάρτησης σε κάθε σημείο αναπαριστά τη φωτεινότητα ή το χρώμα εκείνου του σημείου (Petrou and Petrou 2010).

Η ψηφιακή εικόνα ερμηνεύεται από έναν υπολογιστή ως ένα πλέγμα αριθμητικών δεδομένων, τα pixel (εικονοστοιχεία), ενώ κάθε pixel έχει μια τιμή έντασης φωτεινότητας ή τιμή χρώματος για τις έγχρωμες εικόνες. Υπάρχουν διάφοροι τύποι τέτοιων εικόνων, όπως η δυαδική εικόνα (Binary Image), η οποία αποτελεί μια εικόνα με δύο τιμές, μαύρο και άσπρο ή «0» και «1». Αναφέρεται ως εικόνα 1 bit/pixel επειδή χρειάζεται μόνο ένα δυαδικό ψηφίο για να αναπαραστήσει κάθε pixel (Padmavathi and Thangadurai, 2016). Καθώς απαιτείται μόνο ένα bit για κάθε pixel, χρησιμοποιείται λίγη μνήμη για την αποθήκευση της εικόνας. Αυτή η απλότητα των δυαδικών εικόνων, εξασφαλίζει γρήγορη επεξεργασία. Άλλος ένας τύπος ψηφιακής εικόνας αποτελεί η εικόνα με δείκτες (Indexed Image). Αυτή η εικόνα αποτελείται από έναν πίνακα και έναν πίνακα χρωματικού χάρτη. Ο πίνακας περιέχει τιμές που λειτουργούν ως δείκτες στον χρωματικό χάρτη, ενώ ο χρωματικός χάρτης είναι ένας πίνακας  $m-by-3$  με τιμές στο εύρος  $[0,1]$ , οι οποίες καθορίζουν τα κόκκινα, πράσινα και μπλε (RGB) συστατικά των χρωμάτων. Κάθε τιμή pixel συνδέεται άμεσα με ένα χρώμα του χάρτη (Padmavathi and Thangadurai, 2016). Επιπλέον, υπάρχουν άλλοι δύο κύριοι τύποι ψηφιακής εικόνας, η γκρι εικόνα (Grayscale Image) και η εικόνα RGB (RGB Image). Η γκρι εικόνα είναι μια μονοχρωματική εικόνα, η οποία περιέχει μόνο πληροφορίες φωτεινότητας και καμία πληροφορία χρώματος. Ένα πίξελ σε μια έγχρωμη εικόνα



αποτελείται από τρία βασικά χρώματα, το κόκκινο, το πράσινο και το μπλε, τα οποία αναπαρίστανται στον χώρο RGB. Οι τιμές RGB συνδυάζονται για να δημιουργήσουν μια τρισδιάστατη αναπαράσταση που καθορίζεται από τη φωτεινότητα, τη χρωματικότητα και την απόχρωση (Saravanan, 2010).



Σχήμα 8: : Τα βασικά χρώματα μίας έγχρωμης εικόνας (medium)

Κάθε ένα από τα βασικά χρώματα (Κόκκινο, Πράσινο, Μπλε) μπορεί να καταλάβει 8 bits, ενώ μαζί καταλαμβάνουν συνολικά 24 bits για την αναπαράσταση ενός pixel. Αντίθετα, μια εικόνα αποχρώσεων του γκρι αναπαρίσταται μόνο από την τιμή φωτεινότητας, που καταλαμβάνει 8 bits ανά pixel, με τιμές που κυμαίνονται από 0 (μαύρο) έως 255 (λευκό) (Saravanan, 2010).

Σε γκρι κλίμακα, η ένταση κάθε pixel εκφράζεται με μια τιμή από 0 έως 255, όπου το 0 αντιστοιχεί στο μαύρο χρώμα και το 255 στο λευκό, ενώ σε έγχρωμες εικόνες, η ένταση του κάθε pixel αποθηκεύεται ως συνδυασμός RGB (Κόκκινο, Πράσινο, Μπλε) (Padmavathi and Thangadurai, 2016). Αυτή η αναπαράσταση επιτρέπει την επεξεργασία, την ανάλυση και τη χρήση της εικόνας σε εφαρμογές υπολογιστικής όρασης, όπως στην αναγνώριση αντικειμένων και στη περίπτωση που μελετάται στην αναγνώριση κεραυνών.

### 3.2. Επεξεργασία εικόνας με τη χρήση Τεχνητής Νοημοσύνης

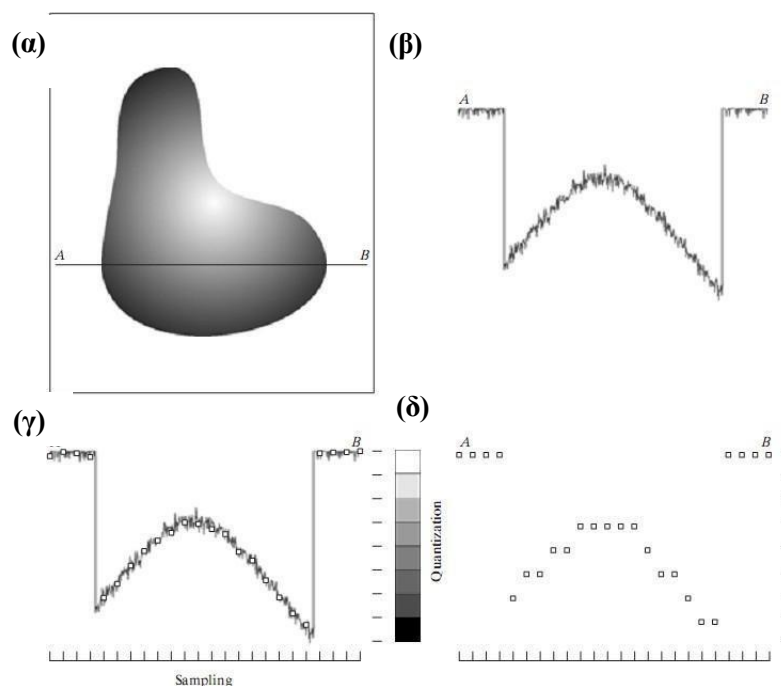
Η επεξεργασία εικόνας με τη χρήση Τεχνητής Νοημοσύνης (TN) αναφέρεται στη χρήση τεχνικών αυτής αλλά και τεχνικών υπολογιστικής όρασης για την ανάλυση, διαχείριση και εξαγωγή χρήσιμων πληροφοριών από ψηφιακές εικόνες αλλά και γενικότερα οπτικών δεδομένων. Αυτή η επεξεργασία χρησιμοποιείται σε διάφορες εφαρμογές, όπως η βελτίωση της ποιότητας και της ανάλυσης της εικόνας, η αφαίρεση του θορύβου για τη βελτίωση της καθαρότητας αλλά και η ανίχνευση και αναγνώριση αντικειμένων ή προσώπων.

Για να είναι δυνατή αυτή η επεξεργασία της εικόνας και η χρήση της από μοντέλα, για την αναγνώριση του αντικειμένου, θα πρέπει να μετατραπεί από ένα αναλογικό σήμα σε ψηφιακό, δηλαδή από ένα συνεχές σήμα σε σήμα με διακριτό πεδίο ορισμού και πεδίο τιμών. Η μετατροπή αυτή συμβαίνει σε δύο στάδια, την δειγματοληψία, η οποία αναφέρεται στη διαδικασία ψηφιοποίησης του πεδίου ορισμού και την κβαντοποίηση, την διαδικασία ψηφιοποίησης του πεδίου τιμών (Pritha and Jeslet, 2015).

Στο στάδιο τη δειγματοληψίας γίνεται ο διαχωρισμός της συνεχούς εικόνας σε ένα διακριτό πλέγμα pixels, όπου κάθε pixel αντιστοιχεί σε μια μικρή περιοχή της εικόνας. Η ανάλυση της εικόνας καθορίζεται από την πυκνότητα των pixels, η οποία μειώνεται, καθώς δομές που έχουν περίπου το μέγεθος της απόστασης δειγματοληψίας και μικρότερες θα χαθούν (Jähne, 2005). Στο στάδιο της κβαντοποίησης γίνεται η μετατροπή έντασης φωτός κάθε pixel σε έναν αριθμό.

Στο Σχήμα 9(α) απεικονίζεται μια συνεχής εικόνα. Στη δειγματοληψία, λαμβάνουμε δείγματα αυτής της εικόνας και συγκεκριμένα στα σημεία κατά μήκος της γραμμής AB, δημιουργώντας ένα σύνολο διακριτών θέσεων, στον άξονα συντεταγμένων  $x$ .

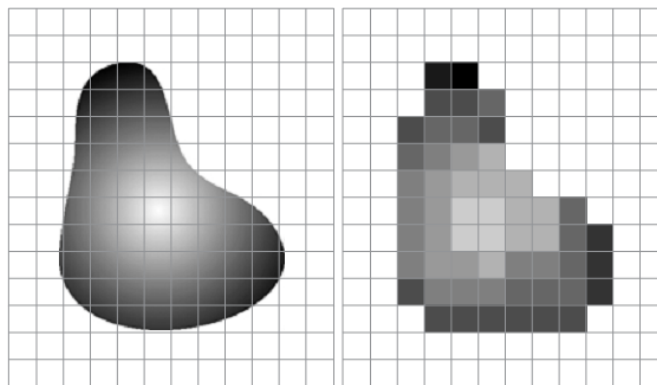
Στο Σχήμα 9(β), διακρίνεται η διαδικασία μετατροπής της συνεχούς εικόνας σε μια συνάρτηση μιας διάστασης (Singhaniya, 2021), η οποία αποτυπώνεται σε μια αναπαράσταση πάνω σε ένα πλέγμα.



Σχήμα 9: Δημιουργία ψηφιακής εικόνας (legendtechz)

Στη συνέχεια, λαμβάνονται δείγματα από αυτή τη γραμμή σε ίσες αποστάσεις, όπως φαίνεται στο Σχήμα 9(γ). Τα δείγματα αυτά αποτελούν το διακριτό σήμα, αλλά οι τιμές τους εξακολουθούν να είναι συνεχείς. Παρουσιάζονται οι γραφικές παραστάσεις της έντασης, κατά μήκος της γραμμής AB της συνεχούς εικόνας. Στη διαδικασία της κβαντοποίησης, γίνεται μετατροπή των συνεχών τιμών έντασης σε διακριτές ψηφιακές τιμές. Το εύρος των τιμών έντασης διαιρείται σε 8 επίπεδα έντασης γκρι, από το μαύρο έως το λευκό, όπως παρουσιάζεται στο σχήμα 9(γ). Οι συνεχείς τιμές έντασης λαμβάνουν την ένταση ενός από τα 8 επίπεδα και τα δείγματα εκχωρούνται στην πλησιέστερη διακριτή τιμή, με βάση τη θέση τους ως προς την κλίμακα (Singhaniya, 2021).

Το αποτέλεσμα της δειγματοληψίας και της κβαντοποίησης, κατά μήκος της γραμμής AB, φαίνεται στο Σχήμα 9(δ). Το τελικό αποτέλεσμα αυτής της διαδικασίας αποτελεί η μετατροπή της γραμμής AB της συνεχούς εικόνας σε μια ψηφιακή γραμμή που αποτελείται από διακριτά σημεία, το καθένα με μια συγκεκριμένη ένταση γκρι. Εφαρμόζοντας αυτή τη διαδικασία γραμμή προς γραμμή, ώστε να καλυφθεί ολόκληρη η εικόνα, παράγεται μια δισδιάστατη ψηφιακή εικόνα, όπως φαίνεται στο παρακάτω σχήμα (Σχήμα 10).



Σχήμα 10: Αποτέλεσμα της δειγματοληψίας και της κβαντοποίησης της εικόνας (legendtechz)

Βέβαια, η ψηφιοποίηση μιας συνεχούς εικόνας συνεπάγεται απώλεια πληροφορίας, καθώς μια συνεχής συνάρτηση μειώνεται σε συνάρτηση σε ένα πλέγμα σημείων (Jähne, 2005).

Η δειγματοληψία σχετίζεται με τα pixel της κάθε εικόνας, με τον συνολικό αριθμό των pixel να υπολογίζεται ως το πηλίκο των συνολικών αριθμών των γραμμών επί των συνολικών αριθμών των στηλών. Επομένως όσο περισσότερα είναι τα δείγματα τόσο περισσότερα θα είναι και τα pixel (Thamali, 2020).

### 3.3. Επεξεργασία εικόνας από υπολογιστή

Μετά τη διαδικασία ψηφιοποίησης, η εικόνα είναι δυνατόν να αναπαρασταθεί ως ένα πλέγμα pixel με τιμές RGB ή γκρι τιμές, ώστε να υποβληθεί σε επεξεργασία.

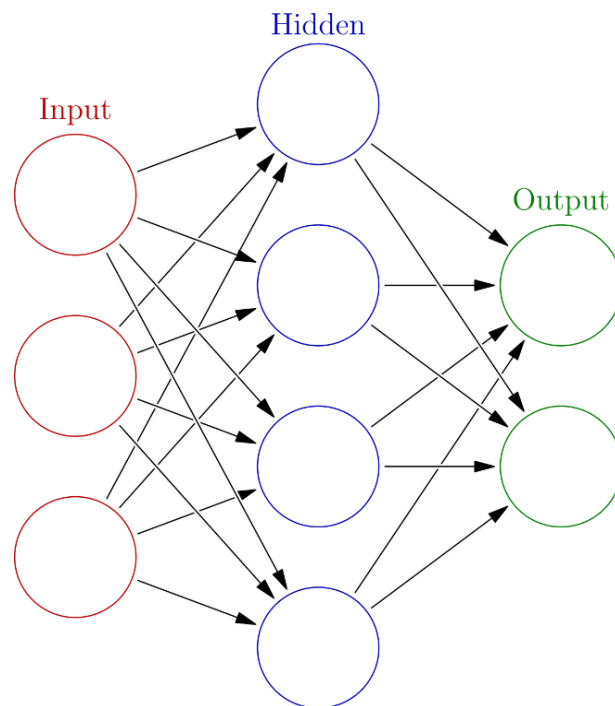
Η επεξεργασία εικόνας με TN βασίζεται σε μοντέλα βαθιάς μάθησης και χρησιμοποιεί προηγμένους αλγορίθμους, νευρωνικά δίκτυα και επεξεργασία δεδομένων για να αναλύσει, να ερμηνεύσει και να επεξεργαστεί ψηφιακές εικόνες.

Για την ανάπτυξη ενός μοντέλου αναγνώρισης ενός αντικειμένου, όπως αυτού του κεραυνού, αρχικά, πρέπει να γίνει συλλογή μεγάλου όγκου δεδομένων, με τα δεδομένα αυτά να αποτελούν αντιπροσωπευτικές εικόνες των αντικειμένων αυτών. Οι εικόνες αυτές μπορεί να χρειαστούν κάποια προ επεξεργασία, όπως αλλαγή μεγέθους, για τη διασφάλιση συνέπειας και την αύξηση της απόδοσης του μοντέλου. Μετά την εισαγωγή των δεδομένων γίνεται η διάσπαση της εικόνας σε pixels και τα δεδομένα είναι έτοιμα να επεξεργαστούν.

### 3.4. Τεχνητά Νευρωνικά Δίκτυα

Τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ) αποτελούν αλγοριθμικά μοντέλα μηχανικής μάθησης εμπνευσμένα από τη δομή και τη λειτουργία του βιολογικού νευρικού συστήματος. Χρησιμοποιούν απλοποιημένα μοντέλα νευρώνων σε σχέση με τα βιολογικά, σχεδιασμένα να επεξεργάζονται πληροφορίες με τρόπο που προσομοιάζει τον ανθρώπινο εγκέφαλο. Η βασική δομή ενός ΤΝΔ περιλαμβάνει το επίπεδο εισόδου, το κρυφό επίπεδο και το επίπεδο εξόδου (Xu et al., 2024) .

Σε αντίθεση με τους παραδοσιακούς ψηφιακούς υπολογισμούς, οι οποίοι βασίζονται στους δυαδικούς αριθμούς (0 και 1), τα νευρωνικά δίκτυα λειτουργούν μέσω της διασύνδεσης των στοιχείων επεξεργασίας. Τα στοιχεία αυτά, προσομοιάζουν τη λειτουργία των νευρώνων στον ανθρώπινο εγκέφαλο, ενώ οι συνδέσεις και τα βάρη τους, καθορίζουν το τελικό αποτέλεσμα (Islam et al., 2019).



Σχήμα 11: Δομή Τεχνητών Νευρωνικών Δικτύων (medium)

### 3.5. Δομή Τεχνητών Νευρωνικών Δικτύων

Τα ΤΝΔ αποτελούνται, όπως αναφέρθηκε, από το επίπεδο εισόδου, το κρυφό επίπεδο και το επίπεδο εξόδου. Το επίπεδο εισόδου (Input Layer) λαμβάνει τα εξωτερικά δεδομένα, τα οποία μπορεί να περιλαμβάνουν δεδομένα, όπως εικόνες. Το κρυφό επίπεδο (Hidden Layer) αποτελεί τον πυρήνα της επεξεργασίας, στο οποίο πραγματοποιείται η κύρια επεξεργασία των δεδομένων, μέσω συνδέσεων με βάρη. Στα κρυφά επίπεδα, κάθε σύνδεση έχει ένα βάρος που καθορίζει τη σημασία κάθε εισόδου και ενισχύει ή εξασθενεί το σήμα. Ο αριθμός των κρυφών επιπέδων και των νευρώνων ποικίλλει, καθώς προσαρμόζεται ανάλογα με την πολυπλοκότητα του δικτύου (Han et al., 2018). Τέλος, το επίπεδο εξόδου (Output Layer), παράγει τα τελικά αποτελέσματα, με βάση τα επεξεργασμένα δεδομένα.

Η διαδικασία μάθησης ενός Τεχνητού Νευρωνικού Δικτύου περιλαμβάνει την προσαρμογή των βαρών μεταξύ των νευρώνων, ώστε να ελαχιστοποιήσει το σφάλμα μεταξύ της προβλεπόμενης τιμής και της σωστής τιμής και να παράγει όσο το δυνατόν ακριβέστερες προβλέψεις (Han et al., 2018).

#### 3.5.1. Λειτουργία του Νευρώνα

Ένα ΤΝΔ μπορεί να περιλαμβάνει παραπάνω από ένα κρυφό επίπεδο, ενώ και ο αριθμός των νευρώνων σε αυτά δεν είναι σταθερός. Όταν αυτοί οι νευρώνες λαμβάνουν εισερχόμενη πληροφορία, αυτή αποθηκεύεται και αθροίζεται. Αυτή η αθροιστική τιμή επεξεργάζεται μέσω μιας συνάρτησης ενεργοποίησης και παράγεται μια έξοδος. Η διαδικασία περιγράφεται από την εξίσωση (1):

$$y = f(a) = \begin{cases} 1, & \sum_{i=1}^n x_i w_i \geq \text{threshold value} \\ 0, & \sum_{i=1}^n x_i w_i \leq \text{threshold value} \end{cases} \quad (1)$$

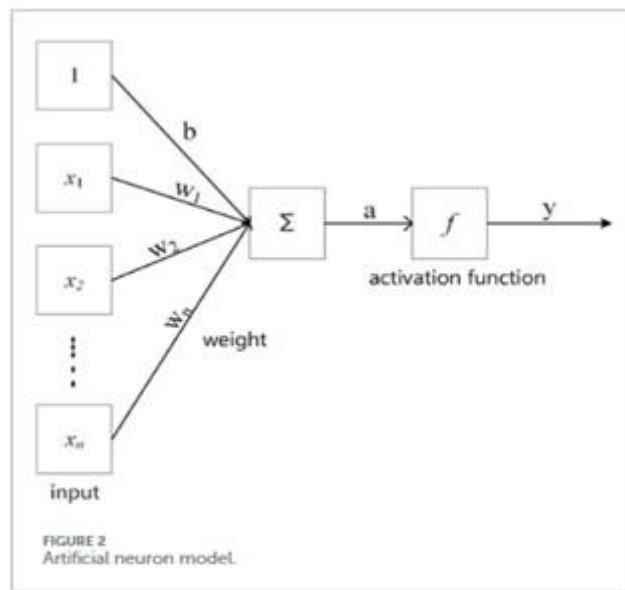
Όπου:

- $X_i$ : Οι είσοδοι.
- $W_i$ : Οι συντελεστές βαρών.

Και το όριο αναφέρεται σε μία τιμή κατωφλίου.

Η έξοδος  $y$  είναι 1 με την ενεργοποίηση του νευρώνα και 0 όταν βρίσκεται σε κατάσταση καταστολής. Επιπλέον, όταν ο νευρώνας προσθέτει τα δεδομένα, χρησιμοποιεί διαφορετικούς συντελεστές βαρών για να υπολογίσει το άθροισμα (Xu et al., 2024).

Η παραπάνω διαδικασία παρουσιάζεται στο Σχήμα 12, στην οποία τα  $W_1, W_2, W_3, W_4$  αποτελούν συντελεστές βαρών κάθε εισερχόμενης πληροφορίας. Ανάλογα με τον τύπο πληροφορίας που επεξεργάζεται το ΤΝΔ, εφαρμόζονται διαφορετικές συναρτήσεις ενεργοποίησης (Yang, 2006).



Σχήμα 12: Μοντέλο Τεχνητού Νευρώνα (Xu et al., 2024)

### 3.5.2. Εκπαίδευση Τεχνητών Νευρωνικών Δικτύων

Τα ΤΝΔ εκπαιδεύονται μέσω δύο βασικών προσεγγίσεων, της μάθησης με επίβλεψη και της μάθησης χωρίς επίβλεψη. Η μάθηση με επίβλεψη απαιτεί δεδομένα εισόδου και εξόδου που φέρουν ετικέτες. Με τη μάθηση με επίβλεψη, το ΤΝΔ εκπαιδεύεται με την καθοδήγηση των ετικετών αυτών, οι οποίες δείχνουν τις σωστές απαντήσεις για κάθε είσοδο. Το μοντέλο «μαθαίνει» τη σχέση μεταξύ εισόδων και εξόδων, ώστε να είναι σε θέση να προβλέψει σωστά τις εξόδους για νέες εισόδους. Αυτή η μάθηση είναι εξαιρετικά χρήσιμη για την επίλυση προβλημάτων σε πραγματικές συνθήκες. Αναλύοντας δεδομένα εκπαίδευσης με ετικέτες, το σύστημα μπορεί να προβλέψει αποτελέσματα για άγνωστα δεδομένα στα οποία δεν έχει εκπαιδευτεί.

Η μάθηση χωρίς επίβλεψη χρησιμοποιεί αλγόριθμους τεχνητής νοημοσύνης για την ανάλυση δεδομένων χωρίς ετικέτες. Οι αλγόριθμοι αυτοί εντοπίζουν κρυφές σχέσεις ή πρότυπα χωρίς την ανθρώπινη παρέμβαση. Ο αλγόριθμος αναλύει τα δεδομένα για να εντοπίσει τα χαρακτηριστικά τους, με ομαδοποίηση ή συσχέτιση (Tishan, 2023). Κάθε μία από αυτές τις μεθόδους έχει διαφορετική εφαρμογή και στόχο, ανάλογα με τη φύση των δεδομένων και το πρόβλημα που καλούνται να λύσουν.

### **3.6. Συνελικτικά Νευρωνικά Δίκτυα**

Τα ΤΝΔ επεξεργάζονται δεδομένα με τη μορφή επίπεδων διανυσμάτων. Αυτό σημαίνει ότι κάθε δεδομένο, όπως εικόνες, πρέπει αρχικά να μετατραπεί σε μονοδιάστατη μορφή. Τα ΤΝΔ χρησιμοποιούνται σε περιπτώσεις, στις οποίες τα δεδομένα δεν παρουσιάζουν χωρική δομή. Βέβαια, για την ανίχνευση αντικειμένων, η επεξεργασία δεδομένων με χωρική πληροφορία είναι απαραίτητη.

Τα Συνελικτικά Νευρωνικά Δίκτυα (ΣΝΔ) έχουν σχεδιαστεί ειδικά για την ανάλυση δεδομένων με χωρική δομή, όπως είναι οι εικόνες. Σε αντίθεση με τις απλές δομές, τα δίκτυα αυτά διατηρούν τη δισδιάστατη ή τρισδιάστατη μορφή των δεδομένων, διευκολύνοντας την ανάλυση των χωρικών δομών.

Τα ΣΝΔ αποτελούν μια αρχιτεκτονική βαθιάς μάθησης, οποία χρησιμοποιείται ευρέως για την επίλυση σύνθετων προβλημάτων (Indolia et al., 2018). Η βασική αρχή λειτουργίας τους περιλαμβάνει τη μετατροπή μιας εικόνας, η οποία αποτελεί το δεδομένου εισόδου, σε έναν χάρτη χαρακτηριστικών, ο οποίος στη συνέχεια περνά από πολλαπλά συνελικτικά και υποδειγματοληπτικά επίπεδα, για να παραχθεί το τελικό αποτέλεσμα. Τα φίλτρα που χρησιμοποιούνται στα συνελικτικά επίπεδα εντοπίζουν τα χαρακτηριστικά της εικόνας, ενώ τα υποδειγματοληπτικά επίπεδα μειώνουν τις διαστάσεις των δεδομένων, διατηρώντας τις περισσότερο σημαντικές πληροφορίες (Haque, 2023).

#### *3.6.1. Αρχιτεκτονική Συνελικτικών Νευρωνικών Δικτύων*

Η αρχιτεκτονική ενός ΣΝΔ περιλαμβάνει τα τρία βασικά μέρη, που παρατηρούνται και στο ΤΝΔ, το επίπεδο εισόδου, τα κρυφά επίπεδα και το επίπεδο εξόδου. Η διαφορά των

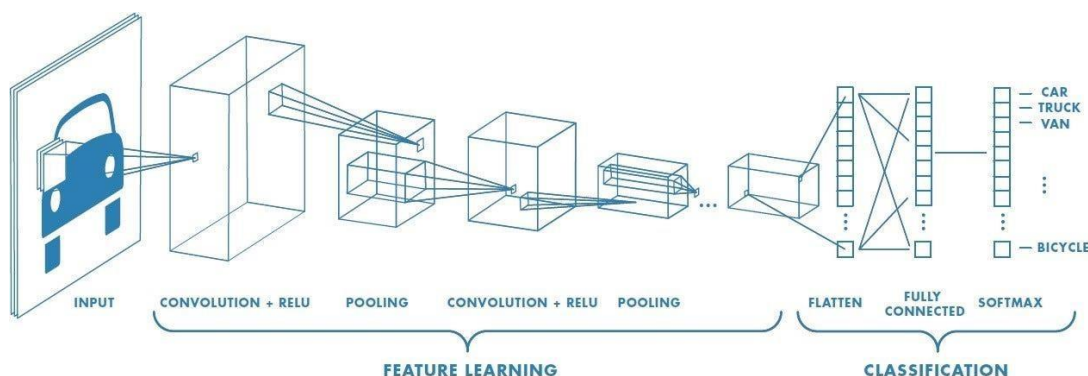


δύο νευρωνικών δικτύων έγκειται στο γεγονός πως τα κρυφά επίπεδα στο ΣΝΔ περιέχουν τα συνελκτικά και υποδειγματοληπτικά επίπεδα, τα οποία εξάγουν και επεξεργάζονται τα χαρακτηριστικά της εικόνας.

### 3.6.2. Λειτουργία Συνελκτικών Νευρωνικών Δικτύων

Η λειτουργία ενός ΣΝΔ ξεκινάει με την εισαγωγή ενός δεδομένου προς επεξεργασία, συνήθως μίας εικόνας, στο επίπεδο εισόδου. Αυτό το επίπεδο προεπεξεργάζεται την εικόνα και τη μεταβιβάζει στα κρυφά επίπεδα. Η προεπεξεργασία κρίνεται απαραίτητη για τη βελτίωση της απόδοσης του μοντέλου, καθώς μειώνει την υπολογιστική πολυπλοκότητα. Τα κρυφά επίπεδα αποτελούν τον πυρήνα του ΣΝΔ και αποτελούνται από πολλαπλά συνελκτικά στρώματα και στρώματα συγκέντρωσης. Τα συνελκτικά επίπεδα εξάγουν χαρακτηριστικά από την εικόνα χρησιμοποιώντας φίλτρα που εντοπίζουν συγκεκριμένα μοτίβα της εικόνας, ενώ μετά τα συνελκτικά επίπεδα ακολουθεί η εφαρμογή της συνάρτησης ενεργοποίησης, η οποία εισάγει τη μη γραμμικότητα στο μοντέλο. Στη συνέχεια, τα στρώματα συγκέντρωσης μειώνουν τη χωρική διάσταση της εικόνας, διατηρώντας τα σημαντικότερα στοιχεία. Ο αριθμός των κρυφών επιπέδων και των φίλτρων σε κάθε επίπεδο μπορεί να προσαρμοστεί για τη βελτιστοποίηση της απόδοσης του δικτύου. Αυτή η διαδικασία συνεχίζεται με την προσθήκη περισσότερων επιπέδων, με αποτέλεσμα το δίκτυο να μαθαίνει ολοένα και πιο σύνθετα χαρακτηριστικά. Τέλος, το επίπεδο εξόδου επεξεργάζεται όλες τις πληροφορίες, ώστε να συνθέσει την τελική πρόβλεψη του δικτύου (Haque, 2023).

Μετά τα συνελκτικά στρώματα αλλά και το διαμοιρασμό των βαρών, εφαρμόζονται πλήρως συνδεδεμένα στρώματα, ώστε να πραγματοποιηθεί η τελική πρόβλεψη, επιτρέποντας στο δίκτυο να λαμβάνει αποφάσεις βασισμένες στα χαρακτηριστικά με τα οποία έχει εκπαιδευτεί (Παλληκάρης, 2023).



Σχήμα 13: Συνελκτικά Νευρωνικά Δίκτυα (Viejo, 2020)

Τα ΣΝΔ βρίσκουν εφαρμογές σε διάφορους τομείς υπολογιστικής όρασης. Ένας σημαντικός τομέας αποτελεί η αναγνώριση αντικειμένων, στην οποία το μοντέλο ανιχνεύει και αναγνωρίζει αντικείμενα συγκεκριμένων κατηγοριών μέσα σε μια εικόνα. Σημαντικό είναι να αναφερθεί πως τα ΣΝΔ είναι ικανά να αναγνωρίζουν αντικείμενα ανεξαρτήτως της θέσης, του μεγέθους ή της περιστροφής τους, και συχνά παρουσιάζουν ανθεκτικότητα σε θορυβώδη δεδομένα (Haque, 2023).

Επιπλέον, η εκπαίδευση των ΣΝΔ γίνεται μόνο μέσω της μάθησης με επίβλεψη, καθώς χρειάζονται δεδομένα με ετικέτες, ώστε να αναγνωρίσουν αντικείμενα μέσω εικόνων ή βίντεο.

### 3.7. Ανίχνευση αντικειμένων με τα Συνελκτικά Νευρωνικά Δίκτυα

Η ανίχνευση αντικειμένων μέσω των αλγορίθμων των Συνελκτικών Νευρωνικών Δικτύων αποτελεί μία εφαρμογή της υπολογιστικής όρασης, η οποία εντοπίζει και προσδιορίζει αντικείμενα μέσα σε μια εικόνα ή ένα βίντεο. Τα αντικείμενα μπορεί να περιλαμβάνουν μεταξύ άλλων φυσικά φαινόμενα, όπως είναι οι κεραυνοί. Κατά τη διαδικασία της εκπαίδευσης, μεγάλος όγκος δεδομένων, σε μορφή εικόνων ή καρέ βίντεο, εισάγεται στο ΣΝΔ μοντέλο, ώστε αυτό να εκπαιδευτεί με σκοπό να είναι ικανό να εντοπίσει τα αντικείμενα βασισμένο στην εμπειρία που έχει αποκτήσει από προηγούμενα δεδομένα. Για να εκπαιδευτεί ένα ΣΝΔ, ώστε να μπορεί να αναγνωρίζει αντικείμενα, αρχικά, προστίθενται ετικέτες που το επισημαίνουν, σε όλα τα δεδομένα που έχουν συλλεχθεί. Αυτά τα δεδομένα επιτρέπουν στο σύστημα να αναγνωρίζει οπτικά χαρακτηριστικά, όπως

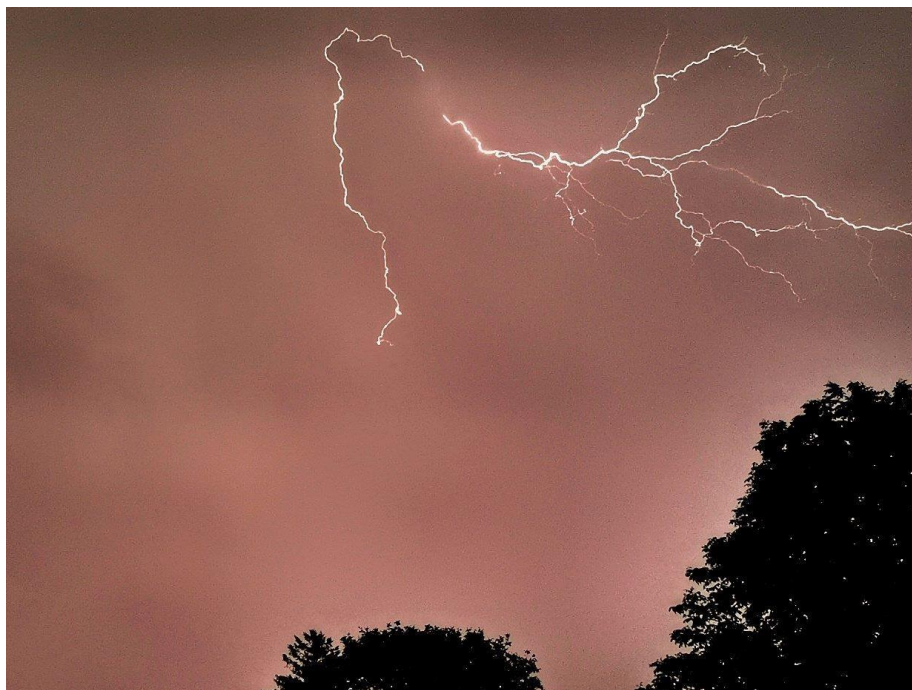
σχήματα, υφές και χρώματα, που διακρίνουν τα αντικείμενα. Ακολουθεί η διαδικασία της εκπαίδευσης με επίβλεψη, η οποία περιλαμβάνει την ανάλυση της εικόνας και στην οποία το μοντέλο προσαρμόζει τις εσωτερικές παραμέτρους του με βάση τη διαφορά μεταξύ των προβλέψεών του και των πραγματικών ετικετών στα δεδομένα εκπαίδευσης. Ένα ξεχωριστό σύνολο επαλήθευσης παρακολουθεί την απόδοση του μοντέλου κατά τη διάρκεια της εκπαίδευσης. Το εκπαιδευμένο μοντέλο είναι έτοιμο για εφαρμογή, σε άγνωστες εικόνες για την παραγωγή προβλέψεων. Όσο περισσότερα δεδομένα αποκτά το σύστημα, τόσο βελτιώνεται η ακρίβεια του στην ανίχνευση αντικειμένων υπό διαφορετικές συνθήκες.

## 4. Μεθοδολογία

Η εκπαίδευση ενός ΣΝΔ στην αναγνώριση κεραυνών και συγκεκριμένα στην ανίχνευση της εμφάνισης κεραυνών είτε σε εικόνες είτε σε πραγματικό χρόνο, είναι χρήσιμη για εφαρμογές όπως η παρακολούθηση καιρικών φαινομένων, η ανάλυση επικινδυνότητας ή η ενίσχυση της ασφάλειας στις επικίνδυνες περιοχές με υψηλή δραστηριότητα καταιγίδας. Ακολουθεί, η μεθοδολογία της δημιουργίας ενός τέτοιου μοντέλου αλλά και στο Κεφάλαιο 5 η υλοποίησή του.

### 4.1. Συλλογή δεδομένων

Για τη δημιουργία ενός μοντέλου που να ανιχνεύει κεραυνούς δημιουργήθηκε ένα ολοκληρωμένο dataset. Έγινε, αρχικά, η δημιουργία ενός dataset με την ονομασία `Dataset_Lightning_Detection`. Αυτό το dataset περιλαμβάνει εικόνες, οι οποίες απεικονίζουν κεραυνούς σε διάφορες καιρικές συνθήκες, τοποθεσίες, συνθήκες φωτισμού, ώρες τις ημέρας και από διαφορετικές οπτικές γωνίες. Το σύνολο των δεδομένων συλλέχθηκε προσεκτικά, ώστε οι εικόνες να είναι όσο το δυνατόν περισσότερο διαφορετικές μεταξύ τους, προσδίδοντας με αυτόν τον τρόπο, η κάθε εικόνα μια νέα πληροφορία ή χαρακτηριστικό ενός κεραυνού στο συνολικό dataset. Απαραίτητη προϋπόθεση για την επιλογή μιας εικόνας αποτέλεσε οι κεραυνοί να είναι ευδιάκριτοι. Για την εκπαίδευση του μοντέλου συλλέχθηκαν, επιπλέον, εικόνες χωρίς την παρουσία κεραυνών, που περιλαμβάνουν σκηνές όπως καθαρούς ή συννεφιασμένους ουρανούς και καιρικά φαινόμενα χωρίς κεραυνούς σε διαφορετικές ώρες τις ημέρας.



Σχήμα 14: Εικόνα από το Dataset\_Training

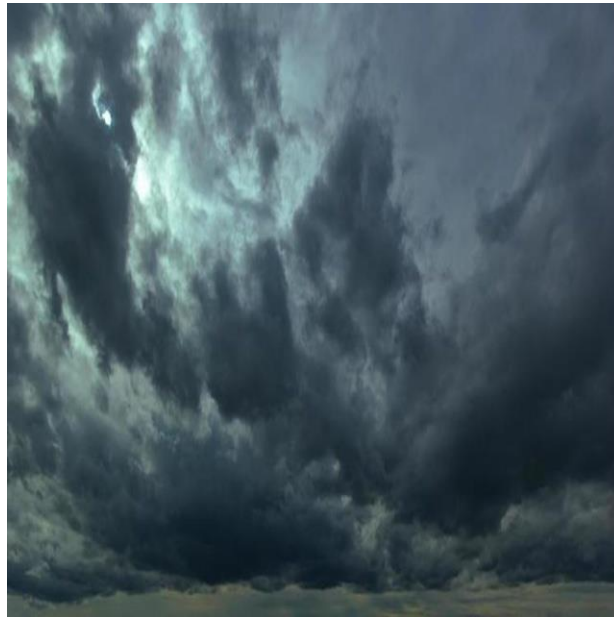
#### 4.1.1. Πηγή δεδομένων

Οι εικόνες συλλέχθηκαν από διάφορες ανοικτές πηγές του διαδικτύου. Έγινε χρήση εικόνων από ιστοσελίδες με δημόσια διαθέσιμες συλλογές εικόνων και συγκεκριμένα από Unsplash και Pixabay, αλλά και από ιστοσελίδες με φωτογραφίες ελεύθερης χρήσης, όπως National Geographic, NASA και NWS. Παρατηρήθηκε πως οι αντιπροσωπευτικές και χρήσιμες, για τη δημιουργία του συγκεκριμένου dataset, εικόνες από κεραυνούς στο διαδίκτυο, ήταν σχετικά περιορισμένες.

#### 4.1.2. Δομή του dataset

Το dataset περιλαμβάνει συνολικά 230 εικόνες, από τις οποίες οι 196 αποτελούν εικόνες που απεικονίζουν κεραυνούς και οι 34 εικόνες χωρίς τη παρουσία αυτού. Το dataset χωρίστηκε σε δύο μέρη με τη κάθε εικόνα να είναι αποθηκευμένη, με το όνομα αρχείου να είναι ένας αριθμός από το 1 έως το 170 για το πρώτο μέρος και val\_1 έως val\_60 για το δεύτερο μέρος. Η ανάλυση των εικόνων ποικίλλει και είναι σε μορφή JPEG. Το dataset έχει χωριστεί σε δύο μέρη, ώστε το ένα μέρος να είναι υπεύθυνο για την εκπαίδευση του μοντέλου και το άλλο για την αξιολόγηση της απόδοσης του μοντέλου σε άγνωστα δεδομένα, σε ποσοστό περίπου 70%-30%.

Συγκεκριμένα, για την εκπαίδευση του μοντέλου χρησιμοποιήθηκαν 170 εικόνες, οι οποίες βρίσκονται στον φάκελο με την ονομασία Dataset\_Training, ενώ για την αξιολόγησή του χρησιμοποιήθηκαν 60 εικόνες στον φάκελο Dataset\_Testing. Στα παραπάνω dataset περιέχονται εικόνες, χωρίς τη παρουσία κεραυνού, ώστε να χρησιμοποιηθούν ως αρνητικά παραδείγματα για την επαλήθευση του μοντέλου.



*Σχήμα 15: Εικόνα που χρησιμοποιήθηκε ως αρνητικό παράδειγμα για την επαλήθευση του μοντέλου*

## **4.2. Επεξεργασία δεδομένων**

Οι εικόνες που συλλέχθηκαν αποτελούσαν ένα σύνολο διαφόρων διαστάσεων και τύπων αρχείων, όπως JPEG, PNG, TIFF και άλλα. Για την ομοιομορφία και τη βελτίωση της κατανόησής τους από το μοντέλο που θα χρησιμοποιηθεί, πραγματοποιήθηκε η μετατροπή τους με τον Κώδικα 1 του Παραρτήματος, σε διαστάσεις 640x640 και η αποθήκευσή τους σε μορφή JPEG.

Λόγω του μεγάλου όγκου των εικόνων και για την αποφυγή διπλότυπων εικόνων που μπορεί να προκύπταν, έγινε χρήση κώδικα για τον υπολογισμό της τιμής hash, ο οποίος παρουσιάζεται στο Παράρτημα (Κώδικας 2). Αυτός ο κώδικας διαβάζει τη τιμή hash των εικόνων, η οποία είναι μια μοναδική αλφαριθμητική αναπαράσταση του περιεχομένου της εικόνας. Η τιμή hash λειτουργεί ως ψηφιακό αποτύπωμα, επομένως εικόνες με διαφορετικά

ονόματα αρχείων αλλά πανομοιότυπο περιεχόμενο θα έχουν την ίδια τιμή hash. Με αυτόν τον τρόπο, εντοπίστηκαν και αφαιρέθηκαν τυχόν διπλότυπα αρχεία.

Ο συνολικός όγκος των δεδομένων ήταν 230 εικόνες, με συνολικό μέγεθος 18.06 MB και μέσο μέγεθος ανά εικόνα 80.41 KB. Ο όγκος δεδομένων για το κάθε dataset, υπολογίστηκε με βάση το Κώδικα 3 του Παραρτήματος. Συγκεκριμένα, για κάθε dataset ο όγκος των δεδομένων ήταν:

Dataset 1: Σύνολο εικόνων: 170

Συνολικό μέγεθος: 13.29 MB

Μέσο μέγεθος ανά εικόνα: 80.04 KB

Dataset 2: Σύνολο εικόνων: 60

Συνολικό μέγεθος: 4.77 MB

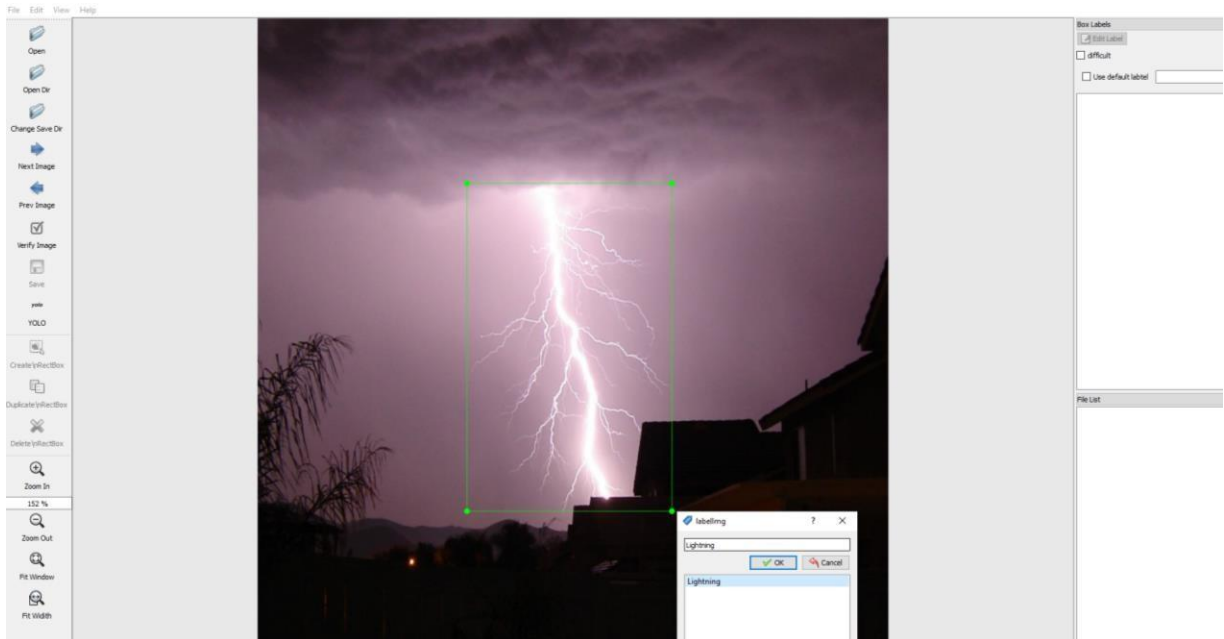
Μέσο μέγεθος ανά εικόνα: 81.45 KB

Στα δύο dataset υπάρχουν εικόνες χωρίς τη παρουσία του κεραυνού και συγκεκριμένα, στο πρώτο dataset υπάρχουν 20 εικόνες και στο δεύτερο 14 εικόνες χωρίς τη παρουσία κάποιας ηλεκτρικής εκκένωσης.

Με την προεπεξεργασία των δεδομένων αυτών, διασφαλίζεται η βέλτιστη απόδοσή τους στο μοντέλο που θα χρησιμοποιηθεί στη συνέχεια, αυξάνοντας με αυτόν τον τρόπο την αποτελεσματικότητά του.

Για την επεξεργασία των εικόνων, ώστε αυτές να μετατραπούν σε δεδομένα χρήσιμα για την εκπαίδευση ενός μοντέλου μηχανικής μάθησης, έγινε χρήση του εργαλείου ανοιχτού κώδικα LabelImg, το οποίο χρησιμοποιείται για την χειροκίνητη επισήμανση (labeling) δεδομένων. Πιο συγκεκριμένα, το LabelImg επιτρέπει την επισήμανση αντικειμένων σε εικόνες, με την οριοθέτηση τους σε πλαίσια οριοθέτησης, δηλαδή σε κουτιά (bounding boxes). Το γεγονός αυτό, το καθιστά ένα πολύ χρήσιμο εργαλείο για μοντέλα υπολογιστικής όρασης. Το LabelImg αποτελεί μια εφαρμογή βασισμένη σε Python, η οποία χρησιμοποιεί Qt για το γραφικό της περιβάλλον, ενώ απαιτείται εγκατάσταση. Τα δεδομένα που παράγονται μπορεί να είναι της μορφής είτε YOLO είτε PASCAL VOC. Το εργαλείο αυτό είναι διαθέσιμο μέσω GitHub και συγκεκριμένα στον σύνδεσμο: <https://github.com/HumanSignal/labelImg>.

Επιπλέον, το LabelImg δεν αναπτύσσεται πλέον ενεργά, καθώς έχει ενσωματωθεί στην κοινότητα του Label Studio. Το Label Studio αποτελεί ένα ανοιχτού κώδικα εργαλείο επισήμανσης δεδομένων, το οποίο δεν υποστηρίζει μόνο εικόνες, αλλά και δεδομένα όπως, κείμενο, ήχο και βίντεο.



Σχήμα 16: Ορισμός ετικετών στην εικόνα από το εργαλείο LabelImg

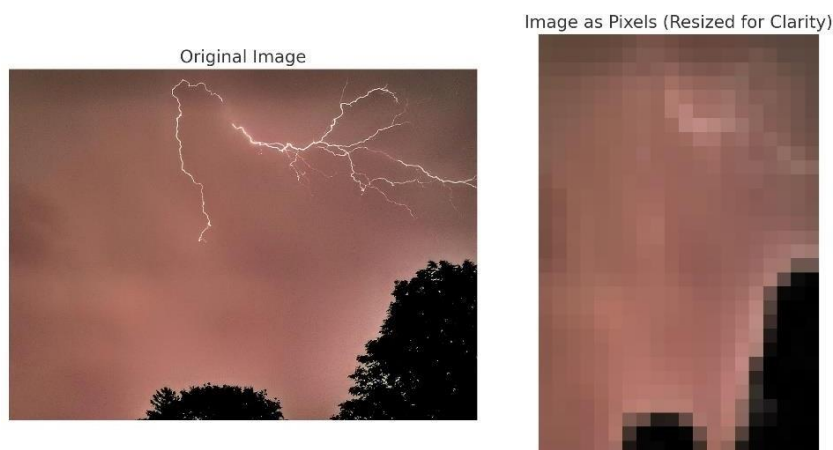
Αρχικά, γίνεται φόρτωση της εικόνας στο LabelImg. Η εικόνα μετατρέπεται σε μια δισδιάστατη αναπαράσταση αριθμών, στην οποία κάθε αριθμός αντιστοιχεί σε ένα εικονοστοιχείο (pixel) της εικόνας. Το εργαλείο αυτό, επιτρέπει, για την επισήμανση των αντικειμένων, τη δημιουργία κουτιών, ώστε να δημιουργούνται πλαίσια που περιβάλλουν τα αντικείμενα ενδιαφέροντος. Η διαδικασία μέσω της οποίας προστίθενται οι ετικέτες (labels) για την αναγνώριση κεραυνών, γίνεται χειροκίνητα και συγκεκριμένα σχεδιάζοντας τα bounding boxes και δίνοντας το όνομα Lightning για τα labels.

Το LabelImg συνοδεύεται από ένα αρχείο που ονομάζεται “predefined\_classes.txt”, το οποίο περιέχει προκαθορισμένες κλάσεις αντικειμένων. Πριν από την εκκίνηση της εφαρμογής, οι προκαθορισμένες αυτές κλάσεις αντικαταστάθηκαν με τη κλάση Lightning, ώστε να εξατομικευτεί η διαδικασία επισήμανσης αποκλειστικά για την ανίχνευση κεραυνών. Τα labels εξάγονται στη μορφή YOLO, η οποία είναι συμβατή με το μοντέλο που θα χρησιμοποιηθεί για την εκπαίδευση.



### 4.3. Επεξεργασία εικόνων και υπολογιστική όραση

Παρουσιάζεται η εικόνα του σχήματος 17, η οποία αποτελεί μέρος του όγκου δεδομένων που συλλέχθηκαν για τη δημιουργία του dataset και δίπλα εμφανίζεται η ίδια εικόνα υπό μορφή pixels και μεγεθυμένη, ώστε να φαίνεται ξεκάθαρα η δομή των pixels.



Σχήμα 17: Τα pixels εικόνας του dataset (Δημιουργήθηκε με τη χρήση τεχνητής νοημοσύνης με το εργαλείο δημιουργίας εικόνων DALL-E)

Κάθε pixel έχει τιμές RGB (Κόκκινο, Πράσινο, Μπλε) που ορίζουν το χρώμα του. Η εικόνα στα δεξιά είναι μια πολύ χαμηλής ανάλυσης αναπαράσταση της αρχικής εικόνας, με ανάλυση 20x30 pixels, για να γίνει κατανοητό πώς αποθηκεύονται οι πληροφορίες της εικόνας υπό μορφή pixel. Με αυτή τη μορφή, η εικόνα εισάγεται σε μοντέλα και γίνεται η ανάλυση και ο εντοπισμός χαρακτηριστικών. Η εικόνα του σχήματος αποτελεί ένα αντιπροσωπευτικό δείγμα ενός κεραυνού, καθώς ο κεραυνός είναι ευδιάκριτος, ενώ ο έντονος φωτισμός του κεραυνού σε αντίθεση με το σκούρο φόντο δημιουργεί ξεκάθαρα περιγράμματα που είναι εύκολο να εντοπιστούν. Το σχήμα του συγκεκριμένου κεραυνού, που αποτελείται από γραμμές και διακλαδώσεις, είναι ιδανικό, καθώς το μοντέλο μπορεί να το μάθει ως χαρακτηριστικό του. Το καθαρό φόντο της εικόνας, με ελάχιστες παρεμβολές από άλλα αντικείμενα, καθιστά την εικόνα κατάλληλη για εκπαίδευση μοντέλων. Επιπλέον, το χρώμα του ουρανού επιτρέπει στο μοντέλο να εκπαιδευτεί σε διάφορες συνθήκες φωτισμού.

Κάθε pixel αποτελεί ένα δεδομένο που εισάγεται στο μοντέλο, το οποίο επεξεργάζεται την εικόνα ως συνδυασμό pixels. Στην εικόνα του σχήματος τα pixels που

ανήκουν στον κεραυνό έχουν πιο φωτεινές τιμές, ενώ τα pixels του ουρανού έχουν πιο σκούρες ή πορτοκαλί τιμές.

#### 4.4. YOLOv5

Το YOLO (You Only Look Once) είναι ένα πολύ δημοφιλές μοντέλο βαθιάς μάθησης για την ανίχνευση αντικειμένων σε πραγματικό χρόνο. Θεωρείται μοντέλο State-of-The-Art (SOTA) σε πολλές εφαρμογές, ενώ αποτελεί μια one-stage detector αρχιτεκτονική. Σε αντίθεση με τα two-stage μοντέλα (όπως το Faster R-CNN), το YOLOv5 κάνει τις προβλέψεις σε μία φάση, συνδυάζοντας τη δημιουργία πλαισίων (bounding boxes) αλλά και την ταξινόμηση αντικειμένων ταυτόχρονα. Αυτά τα πλαίσια υπολογίζονται με βάση την πιθανότητα που τους αποδίδεται. Ο όρος "Look Once" (Κοίτα Μόνο Μια Φορά) υποδηλώνει ότι οι προβλέψεις πραγματοποιούνται με μία μόνο προώθηση (forward propagation) μέσα από το νευρωνικό δίκτυο. Στη συνέχεια, εφαρμόζεται η διαδικασία "Non-Max Suppression", ώστε να διασφαλιστεί ότι κάθε αντικείμενο εντοπίζεται μόνο μία φορά (Sithik, 2023). Το χαρακτηριστικό αυτό, το καθιστά εξαιρετικά γρήγορο και ιδανικό για εφαρμογές σε πραγματικό χρόνο.

Είναι προσβάσιμο μέσω του Github και συγκεκριμένα του URL: <https://github.com/ultralytics/yolov5>. Δίνεται η επιλογή να χρησιμοποιηθεί σε διάφορα περιβάλλοντα, όπως το Colab, το Kaggle, να τρέξει μέσω Gradient και άλλα.

Ακόμη, η εκπαίδευση του YOLOv5 δύναται να πραγματοποιηθεί τοπικά, είτε με τη χρήση της κεντρικής μονάδας επεξεργασίας (CPU) είτε, αν είναι διαθέσιμη, μέσω της μονάδας επεξεργασίας γραφικών (GPU), βέβαια με σημαντικές διαφορές στην απόδοση. Με τη χρήση της GPU επιταχύνεται σημαντικά η εκπαίδευση του μοντέλου, καθώς γίνεται παράλληλη επεξεργασία των δεδομένων, σε αντίθεση με τη CPU, στην οποία η επεξεργασία των δεδομένων εκτελείται σειριακά (Sarrel, 2024).

Για την εκτέλεση του YOLOv5 είναι απαραίτητη η εγκατάσταση του Git και της Python. Το αποθετήριο του YOLOv5 μπορεί να κλωνοποιηθεί μέσω της εντολής:

```
git clone https://github.com/ultralytics/yolov5 # clone
```

```
cd yolov5
```

```
pip install -r requirements.txt # install
```

Η τελευταία εντολή διασφαλίζει ότι θα εγκατασταθούν όλες οι απαιτήσεις, όπως PyTorch, OpenCV αλλά και άλλες βιβλιοθήκες που απαιτούνται για την εκπαίδευση και την ανίχνευση αντικειμένων. Η εκπαίδευση του μοντέλου στην αναγνώριση κεραυνών, γίνεται μέσω του περιβάλλοντος Google Colab, καθώς μέσω αυτού προσφέρεται δωρεάν πρόσβαση σε GPU. Μετά την ολοκλήρωση αυτών των βημάτων, το YOLOv5 είναι έτοιμο για εκτέλεση και χρήση στο περιβάλλον αυτό.

Με το YOLOv5 η εικόνα που θα αναλυθεί υποβάλλεται πρώτα σε επεξεργασία μέσω ενός επιπέδου εισόδου και στη συνέχεια αποστέλλεται στο backbone για εξαγωγή χαρακτηριστικών. Το backbone δημιουργεί χάρτες χαρακτηριστικών (feature maps) σε διαφορετικές κλίμακες, οι οποίοι συνδυάζονται μέσω του δικτύου σύντηξης χαρακτηριστικών (Neck). Από αυτή τη διαδικασία προκύπτουν τρεις χάρτες χαρακτηριστικών, P3, P4 και P5, με διαστάσεις  $80 \times 80$ ,  $40 \times 40$  και  $20 \times 20$ , αντίστοιχα, που χρησιμοποιούνται για την ανίχνευση μικρών, μεσαίων και μεγάλων αντικειμένων στην εικόνα (Liu et al., 2022).

Αυτοί οι τρεις χάρτες μεταφέρονται στο prediction head, όπου γίνεται υπολογισμός της εμπιστοσύνης και παλινδρόμηση των πλαισίων περιγράμματος για κάθε pixel, χρησιμοποιώντας προκαθορισμένα anchors. Το αποτέλεσμα είναι ένας πολυδιάστατος πίνακας (BBboxes), ο οποίος περιέχει πληροφορίες όπως την κατηγορία του αντικειμένου, την εμπιστοσύνη της κατηγορίας, τις συντεταγμένες του πλαισίου, το πλάτος και το ύψος.

Στη συνέχεια, εφαρμόζονται φίλτρα με τη χρήση ορίων για την απομάκρυνση περιττών δεδομένων και εκτελείται η διαδικασία μη μέγιστης καταστολής (NMS) για την τελική επιλογή των αποτελεσμάτων. Η διαδικασία που μετατρέπει την εικόνα εισόδου σε BBboxes ονομάζεται διαδικασία inference και παρουσιάζεται γραφικά στο Σχήμα 18, ενώ η εφαρμογή των φίλτρων και της NMS αναφέρεται ως μετα-επεξεργασία (post-processing), η οποία δεν περιλαμβάνει καμία αλλαγή στη δομή του δικτύου.

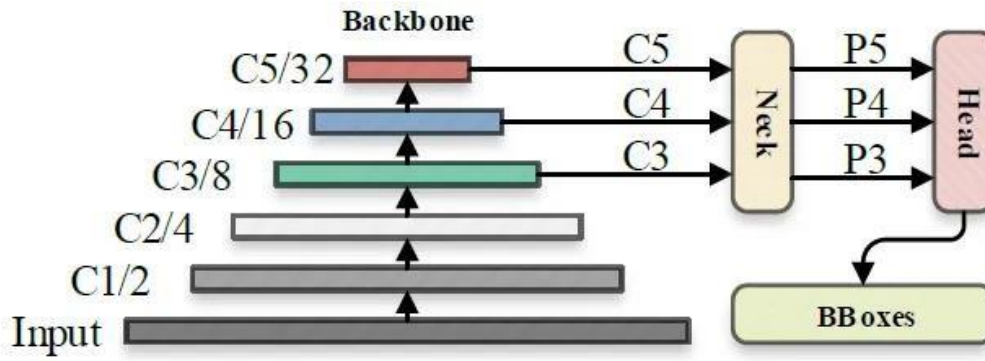
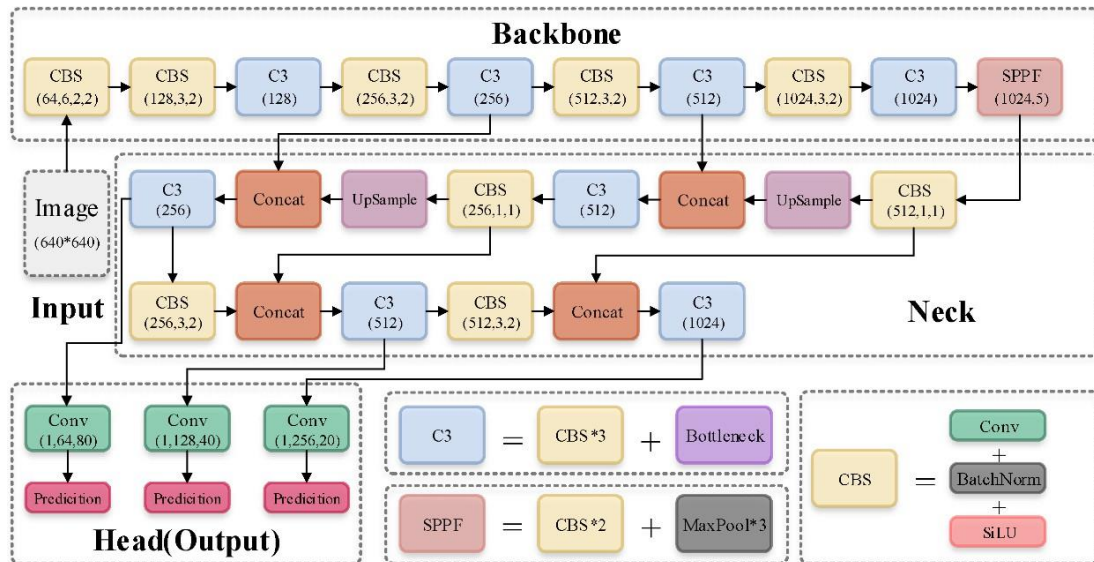


Figure 1. The default inference flowchart of YOLOv5.

Σχήμα 18: Το inference flowchart του YOLOv5 (Tsang, 2023)

#### 4.4.1. YOLOv5: Backbone

Το YOLOv5 χρησιμοποιεί το Cross Stage Partial Network (CSPNet) ενσωματωμένο στο Darknet, δημιουργώντας το CSPDarknet ως backbone. Το CSPNet αντιμετωπίζει το πρόβλημα της επαναλαμβανόμενης πληροφορίας gradient σε backbones μεγάλης κλίμακας, ενώ ενσωματώνει αυτές τις αλλαγές στον χάρτη χαρακτηριστικών. Με αυτόν τον τρόπο, μειώνονται τόσο οι παράμετροι όσο και οι FLOPS (Floating-Point Operations Per Second) του μοντέλου (Xu et al., 2021).

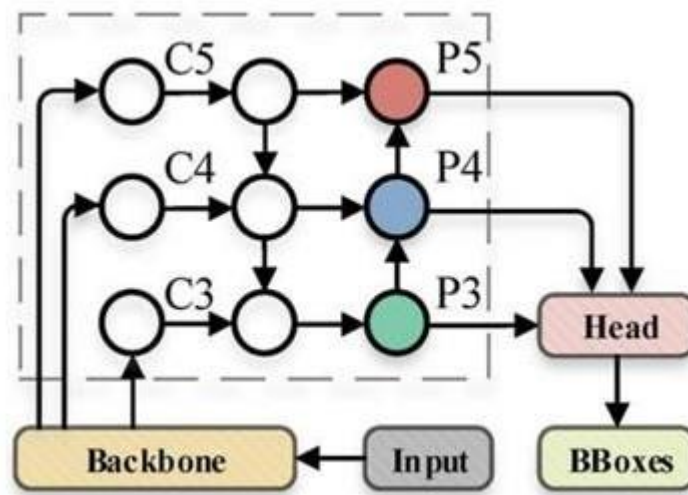


Σχήμα 19: Η δομή δικτύου του YOLOv5 (Tsang, 2023)

Αυτή η προσέγγιση διασφαλίζει την υψηλή ταχύτητα πρόβλεψης και την ακρίβεια, ενώ παράλληλα μειώνει το συνολικό μέγεθος του μοντέλου, το οποίο στο πλαίσιο ανίχνευσης είναι ιδιαίτερα σημαντικό.

#### 4.4.2. YOLOv5: Neck

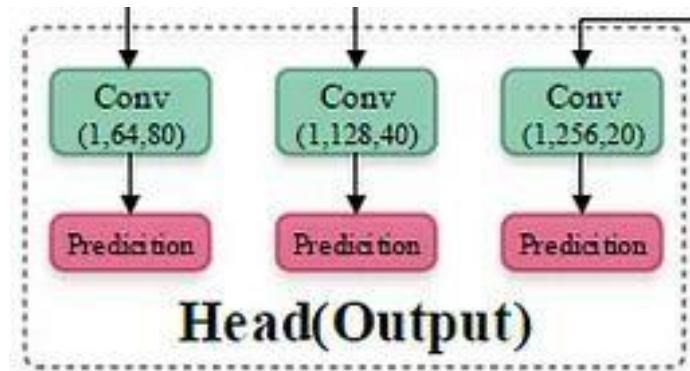
Στο παρακάτω σχήμα (Σχήμα 20) διακρίνεται το Neck του YOLOv5. Η διακεκομμένη γραμμή αναπαριστά την προεπιλεγμένη διαδρομή συγχώνευσης χαρακτηριστικών (feature fusion path) του YOLOv5, ενώ το περιεχόμενο το γραμμών αυτών αποτελεί το Neck του μοντέλου.



Σχήμα 20: YOLOv5: Neck (Tsang, 2023)

#### 4.4.3. YOLOv5: Head

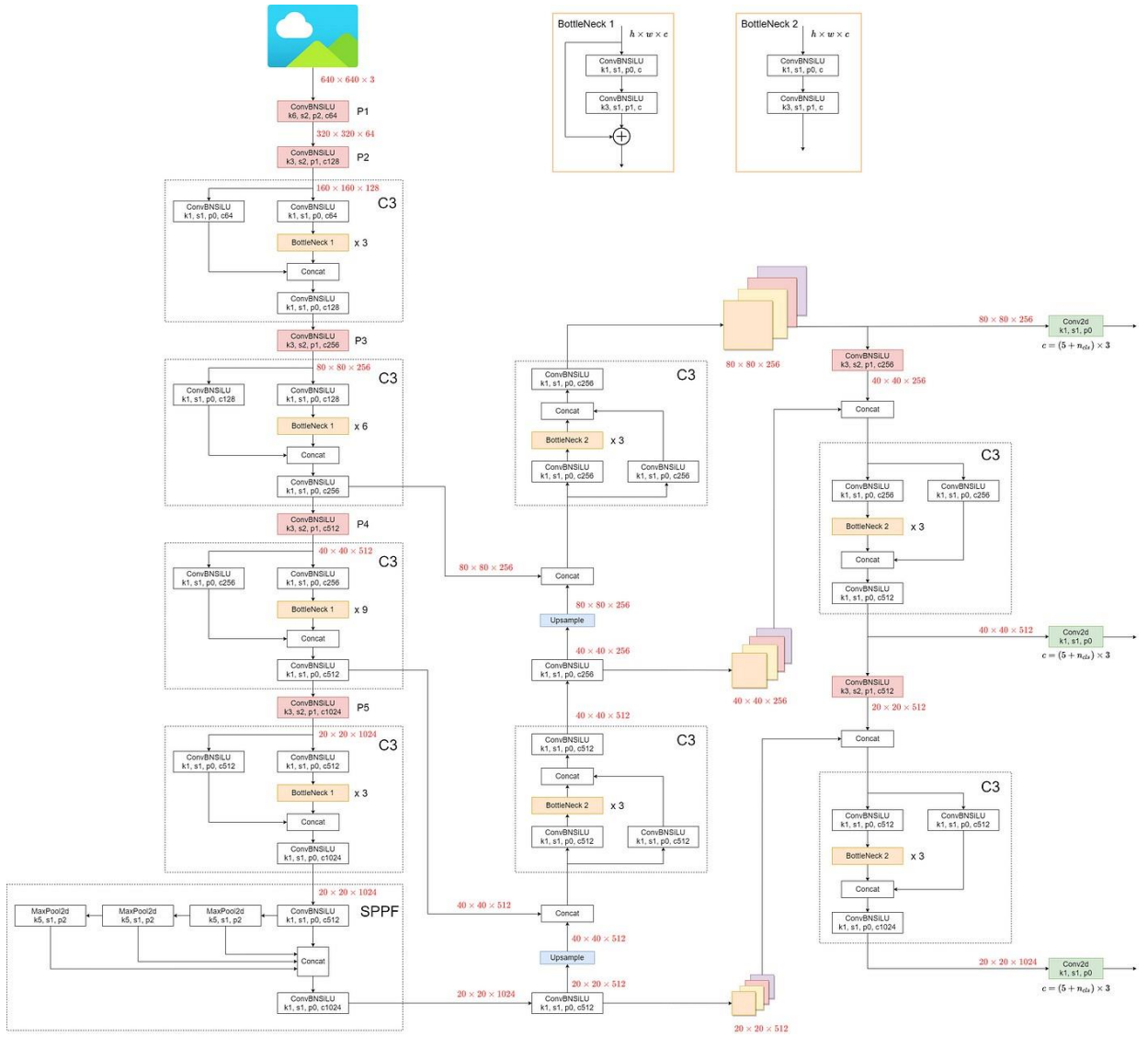
Αυτό το κομμάτι είναι υπεύθυνο για τη δημιουργία προβλέψεων που βασίζονται στα anchor boxes, τα οποία χρησιμοποιούνται για την ανίχνευση αντικειμένων. Οι κύριες λειτουργίες του περιλαμβάνουν τη διαδικασία Non Max Suppression, που εξασφαλίζει την επιλογή του βέλτιστου περιγράμματος, καθώς και τη χρήση συναρτήσεων απώλειας πλαισίου περιγράμματος GIoU (Sithik, 2023). Επομένως, το συγκεκριμένο μέρος του μοντέλου είναι υπεύθυνο για την παραγωγή της τελικής εξόδου (Output).



Σχήμα 21: YOLOv5: Head (Tsang, 2023)

#### 4.4.4. Αρχιτεκτονική του YOLOv5

Το YOLOv5 βασίζεται στις προηγούμενες εκδόσεις του YOLO (YOLOv1–YOLOv4). Οι συνεχείς βελτιώσεις το έχουν καταστήσει κορυφαίο στις επιδόσεις του σε δύο επίσημα σύνολα δεδομένων ανίχνευσης αντικειμένων, το Pascal VOC (Visual Object Classes) και το Microsoft COCO (Common Objects in Context) (Xu et al., 2021). Η αρχιτεκτονική του YOLOv5 παρουσιάζεται στο παρακάτω σχήμα (Σχήμα 22).



Σχήμα 22: Λεπτομερής αρχιτεκτονική του YOLOv5 (Tsang, 2023)

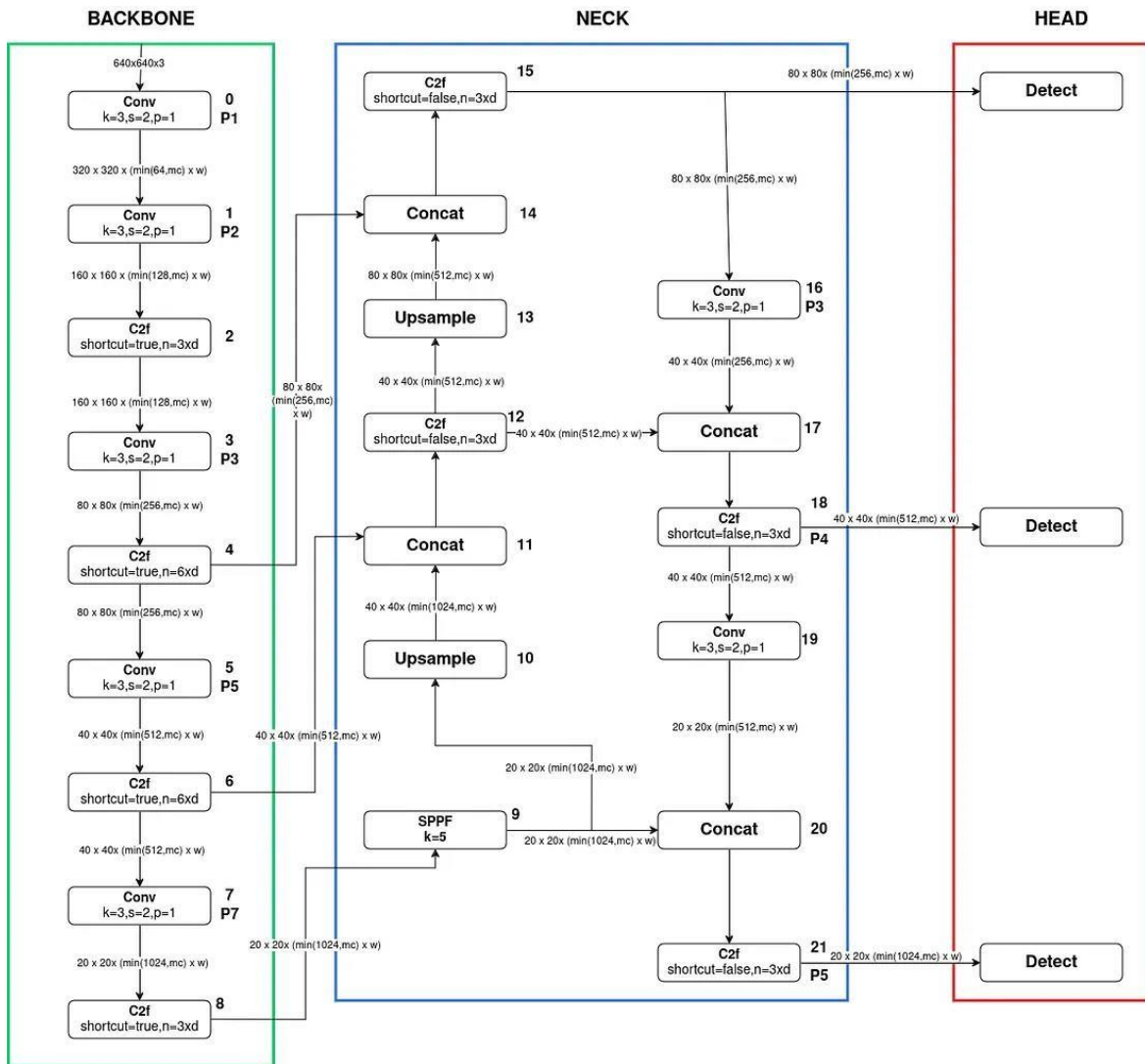
#### 4.5. Δημιουργία του μοντέλου αναγνώρισης κεραυνών στο Yolov5

Η διαδικασία δημιουργίας ενός μοντέλου για την αναγνώριση κεραυνών περιλαμβάνει διάφορα στάδια. Η αρχική εικόνα αναπαρίστανται, όπως αναφέρθηκε, ως πλέγμα pixels, με τις τιμές του να είναι στη κλίμακα RGB. Στο Yolov5 η διαδικασία της προεπεξεργασίας της εικόνας γίνεται αυτόματα πριν την εκπαίδευση του μοντέλου. Η εικόνα προσαρμόζεται στο απαιτούμενο μέγεθος εισόδου και οι τιμές των pixels κανονικοποιούνται σε συγκεκριμένο εύρος.



## 4.6. YOLOv8

Το YOLOv8 αποτελεί μια βελτιωμένη έκδοση του YOLOv5, με τη κύρια διαφορά του, σε σχέση με το YOLOv5, να είναι η χρήση anchor-free detection, το οποίο επιταχύνει τη διαδικασία Non-Maximum Suppression (NMS). Προσφέρει μία πιο αποδοτική αρχιτεκτονική, γρηγορότερη ανίχνευση αλλά και υψηλότερη ακρίβεια (Timilsina, 2024).

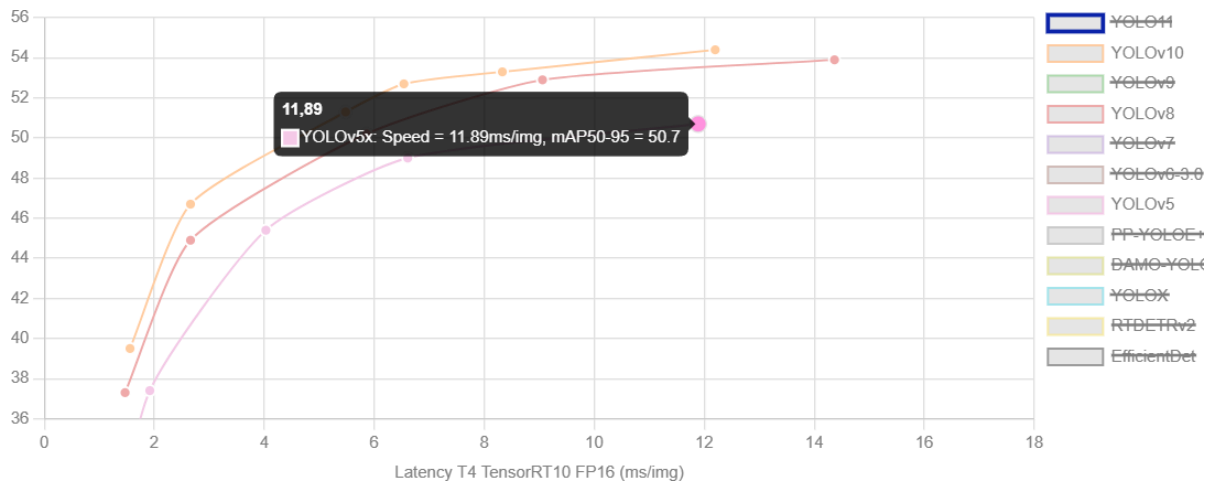


Σχήμα 23: Η αρχιτεκτονική του YOLOv8 (Timilsina, 2024)

Η εξέλιξη της αρχιτεκτονικής του, προσφέρει σημαντικές βελτιώσεις, με το YOLOv8 να υπερτερεί σε σχέση με το YOLOv5, το οποίο αποτελεί προηγούμενη έκδοση, τόσο σε ακρίβεια όσο και σε απόδοση. Στο YOLOv8, η κεφαλή είναι διαχωρισμένη



(Decoupled Head), γεγονός το οποίο προσφέρει καλύτερη πρόβλεψη, σε αντίθεση με το YOLOv5, στο οποίο η κεφαλή είναι ενιαία (Coupled Head), με αποτέλεσμα να προκύπτει υψηλότερη καθυστέρηση (latency). Η υιοθέτηση του C2f module, η μετάβαση σε anchor-free ανίχνευση και η χρήση decoupled head συμβάλλουν στη βελτιστοποίηση της εκμάθησης χαρακτηριστικών και στη μείωση του υπολογιστικού κόστους. Επιπλέον, προηγμένες τεχνικές data augmentation, όπως το Mosaic 2.0 και το Self-Adversarial Training, βελτιώνουν τη γενίκευση του μοντέλου.



Σχήμα 24: Επιδόσεις των μοντέλων YOLOv8 και YOLOv10 (ultralytics)

Παρατηρώντας τον πίνακα, τα YOLOv8 και YOLOv10 παρουσιάζουν καλύτερες επιδόσεις σε σχέση με το YOLOv5, όσον αφορά το mAPval, με χαμηλότερη ή παρόμοια υπολογιστική πολυπλοκότητα (FLOPs).

Model	Size (pixels)	mAP <sub>50-95</sub> <sup>val</sup>	Params (M)	FLOPs (B)
YOLOv5n	640	28.0	2.6	7.7
YOLOv5s	640	37.4	9.1	24.0
YOLOv5m	640	45.4	25.1	64.2
YOLOv5l	640	49.0	53.2	135.0
YOLOv5x	640	50.7	97.2	246.4
YOLOv8n	640	37.3	3.2	8.7
YOLOv8s	640	44.9	11.2	28.6
YOLOv8m	640	50.2	25.9	78.9
YOLOv8l	640	52.9	43.7	165.2
YOLOv8x	640	53.9	68.2	257.8

<b>YOLOv10n</b>	640	39.5	2.3	6.7
<b>YOLOv10s</b>	640	46.8	7.2	21.6
<b>YOLOv10m</b>	640	51.3	15.4	59.1
<b>YOLOv10l</b>	640	53.4	24.4	120.3
<b>YOLOv10x</b>	640	54.4	29.5	160.4

Το YOLOv10 εξαλείφει την ανάγκη για NMS, προσφέροντας καλύτερη ακρίβεια και αποδοτικότητα. Οι "n" εκδόσεις έχουν τον χαμηλότερο αριθμό παραμέτρων, ενώ οι "x" εκδόσεις είναι οι πιο βαριές αλλά αποδίδουν το υψηλότερο mAPval. Το YOLOv10-L και YOLOv10-X φαίνεται να προσφέρουν σημαντική ακρίβεια (53.4% και 54.4% αντίστοιχα), με σχετικά χαμηλή υπολογιστική επιβάρυνση. Επιπλέον, τα FLOPs αυξάνονται εκθετικά καθώς το μέγεθος του μοντέλου μεγαλώνει, επηρεάζοντας την ταχύτητα εκτέλεσης σε πραγματικές εφαρμογές.

#### 4.7. Κατηγοριοποίηση των τύπων των μοντέλων YOLO

Οι τύποι του μοντέλου YOLO, αφορούν διαφορετικές εκδόσεις του ίδιου μοντέλου με εύρος μεγεθών και επιδόσεων. Κάθε τύπος διαφέρει σε μέγεθος, ακρίβεια και ταχύτητα επεξεργασίας. Πιο συγκεκριμένα,

**n (Nano):** Το μικρότερο μοντέλο. Είναι ελαφρύ, γρήγορο αλλά με χαμηλότερη ακρίβεια.

**s (Small):** Αποτελεί ένα μικρό μοντέλο. Προσφέρει ισορροπία μεταξύ ταχύτητας και ακρίβειας.

**m (Medium):** Μεσαίο μοντέλο. Παρέχει υψηλότερη ακρίβεια με μικρότερη ταχύτητα.

**l (Large):** Μεγάλο μοντέλο. Προσφέρει την υψηλότερη ακρίβεια, όμως είναι πιο αργό.

**xl (Extra Large):** Πολύ μεγάλο μοντέλο. Δίνει τη μέγιστη ακρίβεια, όμως απαιτεί περισσότερους υπολογιστικούς πόρους (Timilsina, 2024).

#### 4.8. Google Colab

Το Google Colab (Colaboratory) αποτελεί μια cloud-based υπηρεσία του Jupyter Notebook, η οποία επιτρέπει την εκτέλεση Python κώδικα απευθείας από το πρόγραμμα

περιήγησης, χωρίς να απαιτείται καμία εγκατάσταση. Είναι ιδιαίτερα χρήσιμο για εφαρμογές μηχανικής μάθησης, ανάλυσης δεδομένων και εκπαίδευσης, καθώς παρέχει δωρεάν πρόσβαση σε υπολογιστικούς πόρους, όπως κάρτες γραφικών (GPUs) και μονάδες επεξεργασίας Tensor (TPUs). Τα σημειωματάρια Colab, καθώς εκτελούνται στους διακομιστές της Google Cloud προσφέρουν υψηλή υπολογιστική ισχύ, ανεξάρτητα από τις δυνατότητες του συστήματος από το οποίο χρησιμοποιείται (Google Colab).

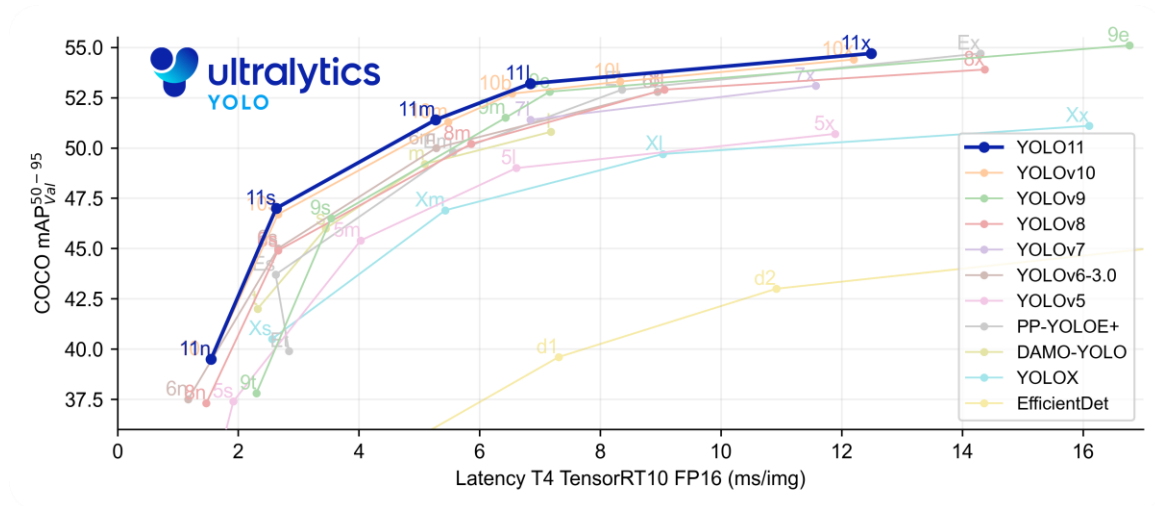
#### **4.9. Αξιολόγηση μοντέλων στην αναγνώριση κεραυνών**

Για την ανάπτυξη ενός μοντέλου αναγνώρισης κεραυνών θα υλοποιηθούν τα YOLOv5 και YOLOv8. Η επιλογή των μοντέλων αυτών, μεταξύ άλλων, βασίστηκε στα χαρακτηριστικά και τις επιδόσεις τους.

Το YOLOv5 (2020) είναι ένα αποδοτικό μοντέλο, ιδιαίτερα γνωστό για την ταχύτητά του και το σχετικά μικρό του μέγεθος, γεγονός που το καθιστά κατάλληλο για ευρύ φάσμα εφαρμογών αλλά και edge συσκευών (Jocher, 2024). Είναι πιο εύχρηστο στην εκπαίδευση και αποτελείται από ολοκληρωμένη τεκμηρίωση, καθώς είναι μία από τις πιο δημοφιλείς εκδόσεις. Αποτελεί ένα αξιόπιστο μοντέλο, όμως δεν αποτελεί τη πιο σύγχρονη έκδοση αλλά αρκετά παλαιότερη.

Το YOLOv8 (2023) παρέχει βελτιωμένες επιδόσεις, σε σύγκριση με τα προηγούμενα μοντέλα και είναι ένα ευέλικτο και σύγχρονο μοντέλο. Όμως, σε σύγκριση με το YOLOv5, απαιτεί περισσότερους υπολογιστικούς πόρους.

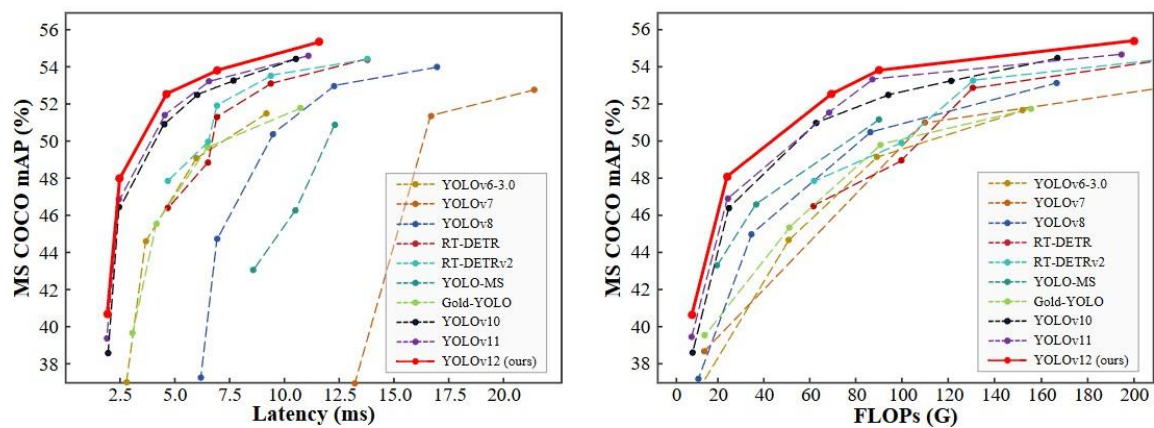
Αναφέρεται, επίσης, το YOLOv10 (2024), καθώς αποτελεί ένα αναβαθμισμένο μοντέλο με υψηλή απόδοση. Βέβαια, αποτελεί νεότερη έκδοση και δεν προσφέρει τόσο τεκμηρίωση όσο οι προηγούμενες εκδόσεις.



Σχήμα 25: Επιδόσεις των μοντέλων YOLO (ultralytics)

#### 4.9.1. Ανάλυση του γραφήματος απόδοσης YOLO μοντέλων

Το παρακάτω γράφημα (Σχήμα 26) συγκρίνει διάφορες εκδόσεις του YOLO καθώς και άλλα μοντέλα ανίχνευσης αντικειμένων με βάση δύο μετρικές, τη COCO mAPval 50-95 (Mean Average Precision), η οποία αναφέρεται στην ακρίβεια του μοντέλου στην ανίχνευση αντικειμένων και τη Latency (T4 TensorRT10 FP16, ms/img), που αναφέρεται στον χρόνο εκτέλεσης ανά εικόνα σε milliseconds.



Σχήμα 26: Σύγκριση μοντέλων σε latency-accuracy (Αριστερά) and FLOPs-accuracy (Δεξιά) trade-offs (github)

Γενικά, όπως παρατηρείται στο γράφημα, το YOLOv11 έχει την καλύτερη απόδοση, πετυχαίνοντας την υψηλότερη ακρίβεια (mAP) σε όλα τα μεγέθη του μοντέλου. Το YOLOv10 έχει επίσης υψηλή απόδοση με μικρότερο latency, σε σύγκριση με το YOLOv8

και YOLOv9. Τα παλαιότερα μοντέλα του YOLO, δηλαδή το YOLOv5, το YOLOv6, το YOLOv7 και το YOLOv8 διαπιστώνεται πως έχουν χαμηλότερη ακρίβεια. Τα υπόλοιπα μοντέλα, όπως το YOLOX, DAMO-YOLO και EfficientDet, παρουσιάζουν σημαντικά χαμηλότερη απόδοση.

Βέβαια, δεν αρκεί μόνο η παραπάνω σύγκριση, ώστε να χαρακτηριστεί ένα YOLO μοντέλο ως κατάλληλο για αναγνώριση αντικειμένων. Υπάρχουν κι άλλοι εξίσου σημαντικοί παράγοντες, οι οποίοι πρέπει να ληφθούν υπόψη. Αυτοί οι παράγοντες αναφέρονται, συγκεκριμένα για την αναγνώριση κεραυνών, στο αν το μοντέλο υποστηρίζει Data Augmentation, αν είναι διαθέσιμο το αρχείο YAML configuration και αν το μοντέλο εκπαιδεύεται με NMS.

Τα Data Augmentation βελτιώνουν τη γενίκευση του μοντέλου, αποτρέπουν το overfitting και αυξάνουν την ποικιλομορφία των δεδομένων εκπαίδευσης. Με το Augmentation δημιουργούνται νέες εκδοχές των εικόνων εκπαίδευσης μέσω μετασχηματισμών, όπως περιστροφή και φωτεινότητα (Ruman, 2023). Είναι πολύ σημαντικό να υποστηρίζεται από το μοντέλο, καθώς αυξάνεται η ποικιλία των δεδομένων χωρίς επιπλέον συλλογή αυτών, ιδιαίτερα αν υπάρχει περιορισμένος όγκος δεδομένων, ενώ βελτιώνει και την ανθεκτικότητα του μοντέλου σε σενάρια, όπως η αλλαγή φωτισμού.

Το αρχείο YAML configuration είναι πολύ σημαντικό, καθώς αυτό αποτελεί το αρχείο που καθορίζει τις ρυθμίσεις του YOLO, όπως η δομή των layers. Τα μοντέλα YOLO, τα οποία δεν έχουν διαθέσιμο το YAML δεν είναι τόσο ευέλικτα για custom training.

Με την εκπαίδευση του μοντέλου με NMS, αυτό κάνει πολλές προβλέψεις για το ίδιο αντικείμενο και μετά φιλτράρει τις λιγότερο αξιόπιστες προβλέψεις, ενώ χωρίς το NMS το YOLO πραγματοποιεί εξ αρχής μόνο μία πρόβλεψη ανά αντικείμενο. Το γεγονός αυτό, καθιστά το μοντέλο πιο γρήγορο και με μεγαλύτερη ακρίβεια, αφού μειώνει τα false positives και τις πλεονάζουσες ανιχνεύσεις, δηλαδή τον εντοπισμό του ίδιου αντικειμένου πολλές φορές με διαφορετικά bounding boxes (Ultralytics).

## 5. Υλοποίηση μοντέλου

Για την υλοποίηση του μοντέλου για την αναγνώριση κεραυνών σε καταιγίδα, αρχικά γίνεται η προσθήκη ετικετών στο `Dataset_Lightning_Detection`, με τη διαδικασία που περιεγράφηκε παραπάνω, με τις ετικέτες να εξάγονται στη μορφή YOLO, η οποία είναι συμβατή με το μοντέλο που θα χρησιμοποιηθεί για την εκπαίδευση, το YOLOv5.



*Σχήμα 27: To Dataset\_ Training του φακέλου Dataset\_Lightning\_Detection*

Το dataset είναι χωρισμένο σε δύο μέρη, το ένα μέρος είναι υπεύθυνο για την εκπαίδευση του μοντέλου και το άλλο για την αξιολόγηση της απόδοσης του σε άγνωστα

δεδομένα. Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκαν 170 εικόνες, οι οποίες βρίσκονται στον φάκελο με την ονομασία Dataset\_Training, ενώ για την αξιολόγησή του χρησιμοποιήθηκαν 60 εικόνες στον φάκελο Dataset\_Testing.



*Σχήμα 28: Το Dataset\_Testing του φακέλου Dataset\_Lightning\_Detection*

Οι ετικέτες που δημιουργήθηκαν από το Labellmg, αποθηκεύτηκαν σε μορφή YOLO. Συγκεκριμένα, για τη παρακάτω φωτογραφία (Σχήμα 29), δημιουργήθηκε μία ετικέτα σε αρχείο .txt, η οποία είναι:

0 0.535156 0.633594 0.320312 0.448437



*Σχήμα 29: Εικόνα από το Dataset\_Testing*

Συγκεκριμένα:

- 0: Ο αριθμός 0 αναφέρεται στην κλάση του αντικειμένου που έχει επισημανθεί, του κεραυνού
- 0.535156 και 0.633594: Αποτελούν τις συντεταγμένες του κέντρου του πλαισίου (bounding box) (x, y), κανονικοποιημένες ως προς το πλάτος και το ύψος της εικόνας.
- 0.320312 και 0.448437: Αποτελούν το πλάτος και το ύψος του πλαισίου, επίσης κανονικοποιημένα.

Όσον αφορά τη κανονικοποίηση, οι τιμές x, y, πλάτος και ύψος κυμαίνονται από 0 έως 1, καθώς είναι διαιρεμένες με το πλάτος και το ύψος της εικόνας. Η εικόνα έχει διαστάσεις 640x640 pixels, οπότε οι πραγματικές συντεταγμένες είναι:

- $x\_center = 0.535156 * 640$
- $y\_center = 0.633594 * 640$
- $box\_width = 0.320312 * 640$
- $box\_height = 0.448437 * 640$



Για τις εικόνες που δεν περιέχουν κεραυνούς, το αρχείο .txt είναι κενό.

## 5.1. Προετοιμασία δεδομένων για το YOLOv5

Οι εικόνες από το dataset αποστέλλονται προς συμπιεσμένο φάκελο (μορφή .zip). Για λόγους απλοποίησης ο φάκελος Dataset\_Training μετονομάζεται σε train και ο φάκελος Dataset\_Testing σε val, καθώς χρησιμοποιείται για την επαλήθευση του μοντέλου (validation). Τα ονόματα των εικόνων του dataset πρέπει να είναι ίδια με τα αντίστοιχα ονόματα των ετικετών.

## 5.2. Εκπαίδευση του μοντέλου

### 5.2.1. YOLOv5

Από το Github και μέσω του περιβάλλοντος του Colab, ξεκινάει η διαδικασία εκπαίδευσης του μοντέλου YoloV5 για την αναγνώριση κεραυνών. Για την εκπαίδευση του μοντέλου, αρχικά, γίνεται σύνδεση με τη Backend Python 3 Google Compute Engine με τη GPU του περιβάλλοντος του Colab, η οποία είναι η Tesla T4 με περίπου 15GB (15095 MiB) μνήμης και επιλέγεται να τρέξει η εντολή Setup, ώστε να γίνει το Clone του GitHub repository, να εγκατασταθούν τα dependencies και να γίνει έλεγχος για PyTorch και GPU. Μόλις τρέξει αυτή η εντολή δημιουργεί στον κατάλογο με τα αρχεία έναν φάκελο με την ονομασία yolov5, στον οποίο θα φορτώσουμε τα δεδομένα για την εκπαίδευση του μοντέλου στην αναγνώριση κεραυνών. Η εντολή Setup:

```
!git clone https://github.com/ultralytics/yolov5 # clone
```

```
%cd yolov5
```

```
%pip install -qr requirements.txt comet_ml # install
```

```
import torch
```

```
import utils
```

```
display = utils.notebook_init() # checks
```

Ο συμπιεσμένος φάκελος που δημιουργήσαμε προηγουμένως εισάγεται στον κατάλογο με τα αρχεία με την ονομασία `train_data` και αποσυμπιέζεται με την εντολή `!unzip -q ../train_data.zip -d ../`.

Από το φάκελο `data` του `yoloV5`, κατεβάζουμε το αρχείο `coco128.yaml`, το οποίο αποτελεί τη σύνδεση των δεδομένων `train` και `val` με το μοντέλο. Το αρχείο μετονομάζεται σε `custom_data` και διαμορφώνεται, ώστε να περιέχει το `path` προς τα δεδομένα εκπαίδευσης και επαλήθευσης, τον αριθμό των κατηγοριών (`classes`) και τις ετικέτες για τον κεραυνό (`Lightning`), όπως φαίνεται στο παρακάτω σχήμα (Σχήμα 30). Το COCO128 αποτελεί ένα μικρό υποσύνολο 128 εικόνων, του πλήρους dataset COCO (Common Objects in Context). Χρησιμοποιείται για γρήγορες δοκιμές και πειραματισμούς αντί του πλήρους COCO dataset, που περιλαμβάνει πάνω από 118,000 εικόνες.

```
4
5  train: ../train_data/images/train
6  val: ../train_data/images/val
7
8  # Classes
9  names:
10 | 0: Lightning
11
```

*Σχήμα 30: Το αρχείο `custom_data`*

Στη συνέχεια, το αρχείο `custom_data`, μεταφορτώνεται στον κατάλογο με τα αρχεία και συγκεκριμένα στην ίδια θέση στην οποία βρισκόταν το αρχείο `coco128.yaml`.

Ακολουθεί η εκπαίδευση του μοντέλου.

```
# Train YOLOv5s on COCO128 for 3 epochs
!python train.py --img 640 --batch 16 --epochs 200 --data custom_data.yaml --weights yolov5s.pt --cache

... 2025-02-22 19:08:17.586161: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:477] Unable to register cuFFT factory: Attempting to register factory for plugin cu
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1740251297.830016 2786 cuda_dnn.cc:8310] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already been regis
E0000 00:00:1740251297.897484 2786 cuda_blas.cc:1418] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been re
train: weights=yolov5s.pt, cfg=, data=custom_data.yaml, hyp=data/hyps/scratch-low.yaml, epochs=200, batch_size=16, imgsz=640, rect=False, resume=False, nosave=False,
github: up to date with https://github.com/ultralytics/yolov5
YOLOv5 v7.0-398-g5cdad892 Python-3.11.11 torch-2.5.1+cu124 CUDA:0 (Tesla T4, 15095MiB)

hyperparameters: lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=0.05, cls=0.5, cls_pw=1.0, obj=
TensorBoard: Start with 'tensorboard --logdir runs/train', view at http://localhost:6006/
COMET WARNING: Comet credentials have not been set. Comet will default to offline logging. Please set your credentials to enable online logging.
COMET INFO: Using '/content/yolov5/.cometml-runs' path as offline directory. Pass 'offline_directory' parameter into constructor or set the 'COMET_OFFLINE_DIRECTORY' env
Downloading https://github.com/ultralytics/yolov5/releases/download/v7.0/yolov5s.pt to yolov5s.pt...
100% 14.1M/14.1M [00:00<00:00, 53.9MB/s]

Overriding model.yaml nc=80 with nc=1

           from  n  params  module  arguments
0             -1  1      3520  models.common.Conv  [3, 32, 6, 2, 2]
1             -1  1     18560  models.common.Conv  [32, 64, 3, 2]
2             -1  1     18816  models.common.C3    [64, 64, 1]
3             -1  1     73984  models.common.Conv  [64, 128, 3, 2]
4             -1  2    115712  models.common.C3    [128, 128, 2]
5             -1  1    295424  models.common.Conv  [128, 256, 3, 2]
6             -1  3    625152  models.common.C3    [256, 256, 3]
7             -1  1   1180672  models.common.Conv  [256, 512, 3, 2]
8             -1  1   1182720  models.common.C3    [512, 512, 1]
9             -1  1    656896  models.common.SPPF  [512, 512, 5]
10            -1  1    131584  models.common.Conv  [512, 256, 1, 1]
11            -1  1         0  torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
12            [-1, 6] 1         0  models.common.Concat  [1]
13            -1  1   361984  models.common.C3    [512, 256, 1, False]
14            -1  1    33024  models.common.Conv  [256, 128, 1, 1]
15            -1  1         0  torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
```

Σχήμα 31: Η εκπαίδευση του μοντέλου YOLOv5

Με την εντολή της python train.py, τρέχει το script train.py, το οποίο είναι υπεύθυνο για την εκπαίδευση του μοντέλου YOLOv5. Η εντολή --img 64 καθορίζει το μέγεθος των εικόνων εισόδου που θα χρησιμοποιηθεί κατά την εκπαίδευση, με το 640 να σημαίνει πως οι εικόνες θα προσαρμοστούν σε διαστάσεις 640x640 pixel. Ένα μεγαλύτερο μέγεθος εικόνας μπορεί να βελτιώσει την ακρίβεια, όμως με αυτόν το τρόπο αυξάνεται η κατανάλωση μνήμης. Η --batch 16 ορίζει το μέγεθος του batch, δηλαδή πόσες εικόνες θα επεξεργάζεται το μοντέλο ταυτόχρονα και η --epochs 3 καθορίζει τον αριθμό των εποχών (epochs) εκπαίδευσης. Κάθε epoch αντιπροσωπεύει έναν πλήρη κύκλο εκπαίδευσης πάνω σε όλα τα δεδομένα. Το batch size παραμένει ίδιο, αφού ένα πολύ μεγάλο μπορεί batch size να εξαντλήσει τη μνήμη της GPU, ενώ τα epoch θα αυξηθούν σε 200, ώστε το μοντέλο να έχει περισσότερο χρόνο να μάθει.

Έγινε δοκιμή διαφορετικών τιμών για τον αριθμό των epochs, ώστε να καθοριστεί η βέλτιστη επιλογή του για την εκπαίδευση του μοντέλου. Παρατηρήθηκε πως, κατά την εκπαίδευση του μοντέλου σε περισσότερες epochs, ενεργοποιήθηκε ο μηχανισμός EarlyStopping, ο οποίος σταμάτησε πρόωρα την εκπαίδευση, καθώς δεν υπήρχε βελτίωση στις τελευταίες 50 epochs. Συγκεκριμένα, η εκπαίδευση σταμάτησε στην 180<sup>η</sup> για τη μικρή και στην 169<sup>η</sup> epoch για τη μεσαία έκδοση του μοντέλου, στις οποίες καταγράφηκαν τα βέλτιστα αποτελέσματα. Διαπιστώθηκε πως περαιτέρω εκπαίδευση δεν προσέφερε κάποια βελτίωση και ως αποτέλεσμα, με αυτή τη διακοπή, αποτρέπεται το overfitting και εξοικονομούνται υπολογιστικοί πόροι.

Τα προκαθορισμένα βάρη που θα φορτωθούν στο μοντέλο πριν ξεκινήσει η εκπαίδευση, καθορίζονται από την εντολή `--weights yolov5s.pt`. Το `yolov5s.pt` είναι ένα προκαθορισμένο μοντέλο YOLOv5 με μικρό μέγεθος (small) και γρήγορη ταχύτητα. Βέβαια, υπάρχουν και άλλα μοντέλα, όπως `yolov5m.pt` (medium), `yolov5l.pt` (large) ή `yolov5x.pt` (extra-large), για μεγαλύτερη ακρίβεια.

### 5.2.2. Ο κώδικας του `train()`

```

103 def train(hyp, opt, device, callbacks):
104     )
105     for epoch in range(start_epoch, epochs): # epoch -----
106         callbacks.run("on_train_epoch_start")
107         model.train()
108
109         # Update image weights (optional, single-GPU only)
110         if opt.image_weights:
111             cw = model.class_weights.cpu().numpy() * (1 - maps) ** 2 / nc # class weights
112             iw = labels_to_image_weights(dataset.labels, nc=nc, class_weights=cw) # image weights
113             dataset.indices = random.choices(range(dataset.n), weights=iw, k=dataset.n) # rand weighted idx
114
115         # Update mosaic border (optional)
116         # b = int(random.uniform(0.25 * imgsiz, 0.75 * imgsiz + gs) // gs * gs)
117         # dataset.mosaic_border = [b - imgsiz, -b] # height, width borders
118
119         mloss = torch.zeros(3, device=device) # mean losses
120         if RANK != -1:
121             train_loader.sampler.set_epoch(epoch)
122         pbar = enumerate(train_loader)
123         LOGGER.info("\n" + "%11s" * 7) % ("Epoch", "GPU_mem", "box_loss", "obj_loss", "cls_loss", "Instances", "Size")
124         if RANK in {-1, 0}:
125             pbar = tqdm(pbar, total=nb, bar_format=TQDM_BAR_FORMAT) # progress bar
126         optimizer.zero_grad()
127         for i, (imgs, targets, paths, _) in pbar: # batch -----
128             callbacks.run("on_train_batch_start")
129             ni = i + nb * epoch # number integrated batches (since train start)
130             imgs = imgs.to(device, non_blocking=True).float() / 255 # uint8 to float32, 0-255 to 0.0-1.0
131
132             # Warmup
133             if ni <= nw:
134                 xi = [0, nw] # x interp
135                 # compute_loss.gr = np.interp(ni, xi, [0.0, 1.0]) # iou loss ratio (obj_loss = 1.0 or iou)
136                 accumulate = max(1, np.interp(ni, xi, [1, nbs / batch_size]).round())
137                 for j, x in enumerate(optimizer.param_groups):
138                     # bias lr falls from 0.1 to lr0, all other lrs rise from 0.0 to lr0
139                     x["lr"] = np.interp(ni, xi, [hyp["warmup_bias_lr"] if j == 0 else 0.0, x["initial_lr"] * lf(epoch)])
140                     if "momentum" in x:
141                         x["momentum"] = np.interp(ni, xi, [hyp["warmup_momentum"], hyp["momentum"]])
142
143             # Multi-scale
144             if opt.multi_scale:
145                 sz = random.randrange(int(imgsz * 0.5), int(imgsz * 1.5) + gs) // gs * gs # size
146                 sf = sz / max(imgs.shape[2:]) # scale factor
147                 if sf != 1:
148                     ns = [math.ceil(x * sf / gs) * gs for x in imgs.shape[2:]] # new shape (stretched to gs-multiple)
149                     imgs = nn.functional.interpolate(imgs, size=ns, mode="bilinear", align_corners=False)
150
151             # Forward
152             with torch.cuda.amp.autocast(amp):
153                 pred = model(imgs) # forward
154                 loss, loss_items = compute_loss(pred, targets.to(device)) # loss scaled by batch_size
155                 if RANK != -1:
156                     loss *= WORLD_SIZE # gradient averaged between devices in DDP mode
157                 if opt.quad:
158                     loss *= 4.0
159
160             # Backward
161             scaler.scale(loss).backward()
162
163     
```

```

103 def train(hyp, opt, device, callbacks):
421     scaler.scale(loss).backward()
422
423     # Optimize - https://pytorch.org/docs/master/notes/amp\_examples.html
424     if ni - last_opt_step >= accumulate:
425         scaler.unscale_(optimizer) # unscale gradients
426         torch.nn.utils.clip_grad_norm_(model.parameters(), max_norm=10.0) # clip gradients
427         scaler.step(optimizer) # optimizer.step
428         scaler.update()
429         optimizer.zero_grad()
430         if ema:
431             ema.update(model)
432         last_opt_step = ni
433
434     # Log
435     if RANK in {-1, 0}:
436         mloss = (mloss * i + loss_items) / (i + 1) # update mean losses
437         mem = f"{torch.cuda.memory_reserved() / 1e9 if torch.cuda.is_available() else 0:.3g}G" # (GB)
438         pbar.set_description(
439             ("%11s" * 2 + "%11.4g" * 5)
440             % (f"{epoch}/{epochs - 1}", mem, *mloss, targets.shape[0], imgs.shape[-1]))
441         )
442         callbacks.run("on_train_batch_end", model, ni, imgs, targets, paths, list(mloss))
443         if callbacks.stop_training:
444             return
445     # end batch -----
446
447     # Scheduler
448     lr = [x["lr"] for x in optimizer.param_groups] # for loggers
449     scheduler.step()
450
451     if RANK in {-1, 0}:
452         # mAP
453         callbacks.run("on_train_epoch_end", epoch=epoch)
454         ema.update_attr(model, include=["yaml", "nc", "hyp", "names", "stride", "class_weights"])
455         final_epoch = (epoch + 1 == epochs) or stopper.possible_stop
456         if not noval or final_epoch: # Calculate mAP
457             results, maps, _ = validate.run(
458                 data_dict,
459                 batch_size=batch_size // WORLD_SIZE * 2,
460                 imgsz=imgsz,
461                 half=amp,
462                 model=ema.ema,
463                 single_cls=single_cls,
464                 dataloader=val_loader,
465                 save_dir=save_dir,
466                 plots=False,
467                 callbacks=callbacks,
468                 compute_loss=compute_loss,
469             )
470
471         # Update best mAP
472         fi = fitness(np.array(results).reshape(1, -1)) # weighted combination of [P, R, mAP@.5, mAP@.5-.95]
473         stop = stopper(epoch=epoch, fitness=fi) # early stop check
474         if fi > best_fitness:
475             best_fitness = fi
476         log_vals = list(mloss) + list(results) + lr
477         callbacks.run("on_fit_epoch_end", log_vals, epoch, best_fitness, fi)
478

```



```

479 # Save model
480 if (not nosave) or (final_epoch and not evolve): # if save
481     ckpt = {
482         "epoch": epoch,
483         "best_fitness": best_fitness,
484         "model": deepcopy(de_parallel(model)).half(),
485         "ema": deepcopy(ema.ema).half(),
486         "updates": ema.updates,
487         "optimizer": optimizer.state_dict(),
488         "opt": vars(opt),
489         "git": GIT_INFO, # {remote, branch, commit} if a git repo
490         "date": datetime.now().isoformat(),
491     }
492
493 # Save Last, best and delete
494 torch.save(ckpt, last)
495 if best_fitness == fi:
496     torch.save(ckpt, best)
497 if opt.save_period > 0 and epoch % opt.save_period == 0:
498     torch.save(ckpt, w / f"epoch{epoch}.pt")
499 del ckpt
500 callbacks.run("on_model_save", last, epoch, final_epoch, best_fitness, fi)
501
502 # EarlyStopping
503 if RANK != -1: # if DDP training
504     broadcast_list = [stop if RANK == 0 else None]
505     dist.broadcast_object_list(broadcast_list, 0) # broadcast 'stop' to all ranks
506     if RANK != 0:
507         stop = broadcast_list[0]
508 if stop:
509     break # must break all DDP ranks
510
511 # end epoch -----
512 # end training -----

```

Σχήμα 32: Ο κώδικας του train()

Η εκπαίδευση του μοντέλου YOLOv8 ακολουθεί αντίστοιχη λογική με το YOLOv5, ωστόσο παρουσιάζονται μερικές κύριες διαφορές στο interface του Google Colab. Σε αντίθεση με το YOLOv5, στο οποίο δίνεται η δυνατότητα να προστίθεται το αρχείο custom\_data.yaml μέσα στον αντίστοιχο φάκελο του μοντέλου, στο YOLOv8 το μονοπάτι στο οποίο βρίσκεται το αρχείο, καθορίζεται δυναμικά στο content directory και καλείται από εκεί κατά την εκπαίδευση. Η διαδικασία εκπαίδευσης παραμένει σχετικά η ίδια, με την σημαντική διαφορά, πως δεν υπάρχει άμεση πρόσβαση στο hyps.yaml αρχείο, το οποίο περιέχει τις υπερπαραμέτρους του μοντέλου, όπως learning rate, momentum και weight decay. Αυτό το αρχείο δεν μπορεί να τροποποιηθεί με τον ίδιο τρόπο όπως στο YOLOv5, στο οποίο η επεξεργασία γινόταν απευθείας μέσα στον φάκελο του μοντέλου.

### 5.2.3. YOLOv8

Στο περιβάλλον του Google Colab, πριν ξεκινήσει η διαδικασία της εκπαίδευσης γίνεται έλεγχος της πρόσβασης σε GPU με την εντολή nvidia-smi. Αυτή η εντολή εμφανίζει πληροφορίες σχετικά με τη GPU, όπως το μοντέλο της (Tesla T4), τη θερμοκρασία (34°C), τη χρήση της μνήμης (15GB διαθέσιμα και 0MiB όταν δεν χρησιμοποιείται), την

έκδοση Driver και CUDA (550.54.15 και 12.4 αντίστοιχα), καθώς και τη χρήση της GPU (0% όταν δεν υπάρχει κανένας ενεργός υπολογισμός).

Ακολουθεί η εντολή `import os`, με την οποία εισάγεται η βιβλιοθήκη `os`, η οποία επιτρέπει την αλληλεπίδραση με το λειτουργικό σύστημα. Η `HOME = os.getcwd()`, αποθηκεύει στη μεταβλητή `HOME` τον τρέχοντα κατάλογο εργασίας και με την `print(HOME)` εκτυπώνεται η διαδρομή του τρέχοντος καταλόγου. Και η έξοδος αποτελεί το `/content`.

```
[2] import os
    HOME = os.getcwd()
    print(HOME)
```

Σχήμα 33: Η εντολή `import os`

Στη συνέχεια, γίνεται εγκατάσταση του YOLOv8, όπως φαίνεται στο παρακάτω σχήμα με δύο τρόπους.

```
[4] # Pip install method (recommended)

!pip install ultralytics==8.2.103 -q

from IPython import display
display.clear_output()

import ultralytics
ultralytics.checks()
```

🔄 Ultralytics YOLOv8.2.103 🚀 Python-3.11.11 torch-2.5.1+cu124 CUDA:0 (Tesla T4, 15095MiB)  
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 33.2/112.6 GB disk)

```
[5] from ultralytics import YOLO

    from IPython.display import display, Image
```

Σχήμα 34: Η εγκατάσταση του YOLOv8

Αφού εκτελεστούν οι παραπάνω διαδικασίες, γίνεται μεταφόρτωση του φακέλου `train_data` και η αποσυμπίεση του, ενώ ακολουθεί η εκπαίδευση του μοντέλου.

```
%cd {HOME}

!yolo task=detect mode=train model=yolov8s.pt data=../content/custom_data.yaml epochs=200 imgsz=640 plots=True iou=0.5 patience=50

E0000 00:00:1740315242.849940 8072 cuda_blas.cc:1418] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been re
Overriding model.yaml nc=80 with nc=1

   from  n  params module                    arguments
   ----  -  -
0         -1 1    928 ultralytics.nn.modules.conv.Conv [3, 32, 3, 2]
1         -1 1   18560 ultralytics.nn.modules.conv.Conv [32, 64, 3, 2]
2         -1 1   29056 ultralytics.nn.modules.block.C2f [64, 64, 1, True]
3         -1 1   73984 ultralytics.nn.modules.conv.Conv [64, 128, 3, 2]
4         -1 2   197632 ultralytics.nn.modules.block.C2f [128, 128, 2, True]
5         -1 1   295424 ultralytics.nn.modules.conv.Conv [128, 256, 3, 2]
6         -1 2   788480 ultralytics.nn.modules.block.C2f [256, 256, 2, True]
7         -1 1  1180672 ultralytics.nn.modules.conv.Conv [256, 512, 3, 2]
8         -1 1  1838080 ultralytics.nn.modules.block.C2f [512, 512, 1, True]
9         -1 1   656896 ultralytics.nn.modules.block.SPPF [512, 512, 5]
10        -1 1      0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
11      [-1, 6] 1      0 ultralytics.nn.modules.conv.Concat [1]
12        -1 1   591360 ultralytics.nn.modules.block.C2f [768, 256, 1]
13        -1 1      0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
14      [-1, 4] 1      0 ultralytics.nn.modules.conv.Concat [1]
15        -1 1  1482224 ultralytics.nn.modules.block.C2f [384, 128, 1]
16        -1 1  147712 ultralytics.nn.modules.conv.Conv [128, 128, 3, 2]
17      [-1, 12] 1      0 ultralytics.nn.modules.conv.Concat [1]
18        -1 1   493056 ultralytics.nn.modules.block.C2f [384, 256, 1]
19        -1 1   590336 ultralytics.nn.modules.conv.Conv [256, 256, 3, 2]
20      [-1, 9] 1      0 ultralytics.nn.modules.conv.Concat [1]
21        -1 1  1969152 ultralytics.nn.modules.block.C2f [768, 512, 1]
22      [15, 18, 21] 1  2116435 ultralytics.nn.modules.head.Detect [1, [128, 256, 512]]

Model summary: 225 layers, 11,135,987 parameters, 11,135,971 gradients, 28.6 GFLOPs
```

Σχήμα 35: Η εκπαίδευση του YOLOv8

Από την έξοδο παρατηρείται πως ξεκινάει η εκπαίδευση του YOLOv8 για ανίχνευση αντικειμένων (task=detect) με τις παραμέτρους:

Μοντέλο: yolov8s.pt

Δεδομένα: ../content/custom\_data.yaml

Epochs: 200

Μέγεθος εικόνας (imgsz): 640x640 pixels

plots = True: Ενεργοποίηση αποθήκευσης οπτικοποιήσεων (τα bounding boxes).

iou = 0.5: Καθορίζει το κατώφλι για το NMS (Non-Maximum Suppression), στη περίπτωση που δύο προβλέψεις έχουν IoU πάνω από 0.5, η μία απορρίπτεται. Μία υψηλή τιμή iou, δύναται να μειώσει την επικάλυψη αντικειμένων, όμως μπορεί να οδηγήσει σε χαμένες ανιχνεύσεις, ενώ μία χαμηλότερη τιμή διατηρεί περισσότερες προβλέψεις, αλλά προκύπτουν περισσότερα false positives. Για αυτό επιλέχθηκε μία μέση τιμή, η 0.5.

patience = 50: Με την εντολή αυτή, το μοντέλο θα σταματήσει την εκπαίδευση αν δεν υπάρχει βελτίωση για 50 συνεχόμενα epochs. Το γεγονός αυτό συντελεί στην αποφυγή περιττών επαναλήψεων αν το μοντέλο έχει ήδη φτάσει στο μέγιστο των δυνατοτήτων του και το σημαντικότερο στην αποφυγή της υπερεκπαίδευσης (overtraining).

Περίληψη Μοντέλου:



- 225 layers
- 13.1M παράμετροι
- 28.6 GFLOPs

```
%cd {HOME}

[yolo task=detect mode=train model=yolov8s.pt data=../content/custom_data.yaml epochs=200 imgsz=640 plots=True iou=0.5 patience=50
22      [15, 18, 21] 1 2116435 ultralytics.nn.modules.head.Detect [1, [128, 256, 512]]
Model summary: 225 layers, 11,135,987 parameters, 11,135,971 gradients, 28.6 GFLOPs

Transferred 349/355 items from pretrained weights
TensorBoard: Start with 'tensorboard --logdir runs/detect/train3', view at http://localhost:6006/
Freezing layer 'model.22.dfl.conv.weight'
AMP: running Automatic Mixed Precision (AMP) checks with YOLOv8n...
AMP: checks passed
train: Scanning /content/train_data/labels/train.cache... 170 images, 20 backgrounds, 0 corrupt: 100% 170/170 [00:00<?, ?it/s]
/usr/local/lib/python3.11/dist-packages/ultralytics/data/augment.py:1837: UserWarning: Argument(s) 'quality_lower' are not valid for transform ImageCompression
A.ImageCompression(quality_lower=75, p=0.0),
augmentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01, num_output_channels=3, method='weighted_average'), CLAHE(p=0.01, c
val: Scanning /content/train_data/labels/val.cache... 60 images, 14 backgrounds, 0 corrupt: 100% 60/60 [00:00<?, ?it/s]
Plotting labels to runs/detect/train3/labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr' and 'momentum' automatically...
optimizer: AdamW(lr=0.002, momentum=0.9) with parameter groups 57 weight(decay=0.0), 64 weight(decay=0.0005), 63 bias(decay=0.0)
TensorBoard: model graph visualization added
Image sizes 640 train, 640 val
Using 2 dataloader workers
Logging results to runs/detect/train3
Starting training for 200 epochs...

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
1/200    4.01G    1.82      4.237     1.97      23          640: 100% 11/11 [00:04<00:00, 2.58it/s]
Class   Images  Instances  Box(P  R  mAP50  mAP50-95): 100% 2/2 [00:00<00:00, 2.80it/s]
all      60      96        0.308  0.385  0.236  0.106
```

Σχήμα 36: Η παράμετροι εκπαίδευσης του YOLOv8

Στη συνέχεια, γίνεται φόρτωση των προεκπαιδευμένων βαρών (yolov8s.pt) και η χρήση της AMP, το οποίο μπορεί να επιταχύνει την εκπαίδευση. Φορτώνονται τα δεδομένα, 170 εικόνες για εκπαίδευση και 60 για την επαλήθευση και παρατηρείται πως δεν εμφανίζονται κατεστραμμένες εικόνες (0 corrupt). Γίνεται βελτιστοποίηση υπερπαραμέτρων και ορίζεται αυτόματα ο καλύτερος optimizer (AdamW).

Το Albumentations είναι μια βιβλιοθήκη για τεχνικές data augmentation. Από το log, φαίνεται ότι εφαρμόζονται οι εξής μετασχηματισμοί στις εικόνες εκπαίδευσης: MedianBlur, με το οποίο εφαρμόζεται ένα φίλτρο θόλωσης με πιθανότητα 10% ( $p=0.1$ ), δηλαδή εφαρμόζεται τυχαία στο 10% των εικόνων και τυχαίο μέγεθος kernel μεταξύ 3 και 7. Το Gray, το οποίο μετατρέπει την εικόνα σε γκρι κλίμακα, με πιθανότητα 1% ( $p=0.01$ ) και το CLAHE, το οποίο εφαρμόζει το Contrast Limited Adaptive Histogram Equalization για καλύτερη αντίθεση, επίσης με πιθανότητα 1%. Τα παραπάνω βοηθούν στη καλύτερη γενίκευση του μοντέλου, βελτιώνοντας την ικανότητά του να αναγνωρίζει αντικείμενα υπό διαφορετικές συνθήκες φωτισμού και θορύβου.

Ακολουθεί η έναρξη της διαδικασίας της εκπαίδευσης και θα τελειώσει με την ολοκλήρωση των 200 epochs.

### 5.3. Αξιολόγηση μοντέλου

Για την αξιολόγηση του μοντέλου γίνεται η ανάλυση των απωλειών αλλά και των μετρικών της εκπαίδευσης. Οι μετρικές αυτές είναι:

#### 5.3.1. Απώλειες (*Losses*)

Οι απώλειες υπολογίζουν το πόσο απέχουν οι προβλέψεις του μοντέλου από τις σωστές απαντήσεις κατά τη διάρκεια της εκπαίδευσης. Χαμηλότερες τιμές υποδηλώνουν καλύτερη απόδοση.

- **train/box\_loss και val/box\_loss**

Υπολογίζουν κατά πόσο ταιριάζουν τα πλαίσια (bounding boxes) που προβλέπει το μοντέλο με τα ground truth boxes. Χαμηλότερη τιμή σημαίνει πως τα πλαίσια είναι πιο ακριβή.

- **train/obj\_loss και val/obj\_loss**

Υπολογίζουν κατά πόσο το μοντέλο αναγνωρίζει ότι ένα αντικείμενο υπάρχει ή δεν υπάρχει σε μια περιοχή. Χαμηλότερη τιμή σημαίνει πως το μοντέλο εντοπίζει καλύτερα την παρουσία ή απουσία αντικειμένων.

- **train/cls\_loss και val/cls\_loss**

Υπολογίζουν το σφάλμα στην κατηγοριοποίηση (classification) των αντικειμένων, δηλαδή στο να τα κατατάσσει στη σωστή κατηγορία. Αν είναι μηδενικό, σημαίνει ότι το μοντέλο δεν εκπαιδεύεται για κατηγοριοποίηση ή ότι οι κατηγορίες δεν υπάρχουν σωστά στα δεδομένα.

#### 5.3.2. Μετρικές

Οι μετρικές αξιολογούν την πραγματική απόδοση του μοντέλου, δηλαδή πόσο ακριβής είναι η πρόβλεψη των αντικειμένων κατά την επαλήθευση. Οι μετρικές είναι: η Precision, η οποία υπολογίζει την ακρίβεια των προβλέψεων, η Recall, η οποία υπολογίζει

την πληρότητα των προβλέψεων, η  $mAP_{0.5}$  (Mean Average Precision), η οποία είναι η σημαντικότερη μετρική καθώς αποτελεί το μέσο όρο της ακρίβειας όταν το IoU (Intersection over Union) είναι τουλάχιστον 0.5 (50% overlap). Υπολογίζει πόσο ταιριάζουν τα πλαίσια και αν το μοντέλο προβλέπει σωστά τις κατηγορίες. Τέλος η μετρική  $mAP_{0.5:0.95}$ , στην οποία αξιολογείται η μέση ακρίβεια για πολλαπλά thresholds του IoU, από 0.5 έως 0.95.

## 6. Σύγκριση πειραματικών μοντέλων

Το 1<sup>ο</sup> πειραματικό μοντέλο εκπαιδεύεται σύμφωνα με τις παραμέτρους του Πίνακα 2. Σε αυτό γίνεται χρήση των yolov5s.pt και yolov5m.pt για τον καθορισμό των βαρών. Τα αποτελέσματα του παρατηρούνται στο Σχήμα . Το 2<sup>ο</sup> πειραματικό μοντέλο χρησιμοποιεί τα yolov8s.pt και yolov8m.pt για τον καθορισμό των βαρών. Όσον αφορά τα Data augmentation αυτά είναι τα default του μοντέλου.

Πίνακας 2: Παράμετροι εκπαίδευσης

<b>lr0</b>	0.01	<b>box</b>	0.05	<b>hsv_h</b>	0.015
<b>lrf</b>	0.01	<b>cls</b>	0.5	<b>hsv_s</b>	0.7
<b>momentum</b>	0.937	<b>cls_pw</b>	1.0	<b>hsv_v</b>	0.4
<b>weight_decay</b>	0.0005	<b>obj</b>	1.0	<b>translate</b>	0.1
<b>warmup_epochs</b>	3.0	<b>obj_pw</b>	1.0	<b>scale</b>	0.5
<b>warmup_momentum</b>	0.8	<b>iou_t</b>	0.2	<b>fliplr</b>	0.5
<b>warmup_bias_lr</b>	0.1	<b>anchor_t</b>	4.0	<b>mosaic</b>	1.0

Επιπλέον, στις τελευταίες 10 εποχές όλων των μοντέλων, εφαρμόστηκαν τεχνικές data augmentation, μέσω της βιβλιοθήκης Albumentations για την ενίσχυση των δεδομένων εκπαίδευσης. Οι τεχνικές αυτές περιλάμβαναν τα παρακάτω.

albumentations: Blur(p=0.01, blur\_limit=(3, 7)), MedianBlur(p=0.01, blur\_limit=(3, 7)), ToGray(p=0.01, num\_output\_channels=3, method='weighted\_average'), CLAHE(p=0.01, clip\_limit=(1.0, 4.0), tile\_grid\_size=(8, 8))

Αυτά τα augmentation στοχεύουν στη βελτίωση της ικανότητας του μοντέλου να αναγνωρίζει αντικείμενα υπό διαφορετικές παραμέτρους και στη καλύτερη γενίκευση σε άγνωστα δεδομένα.

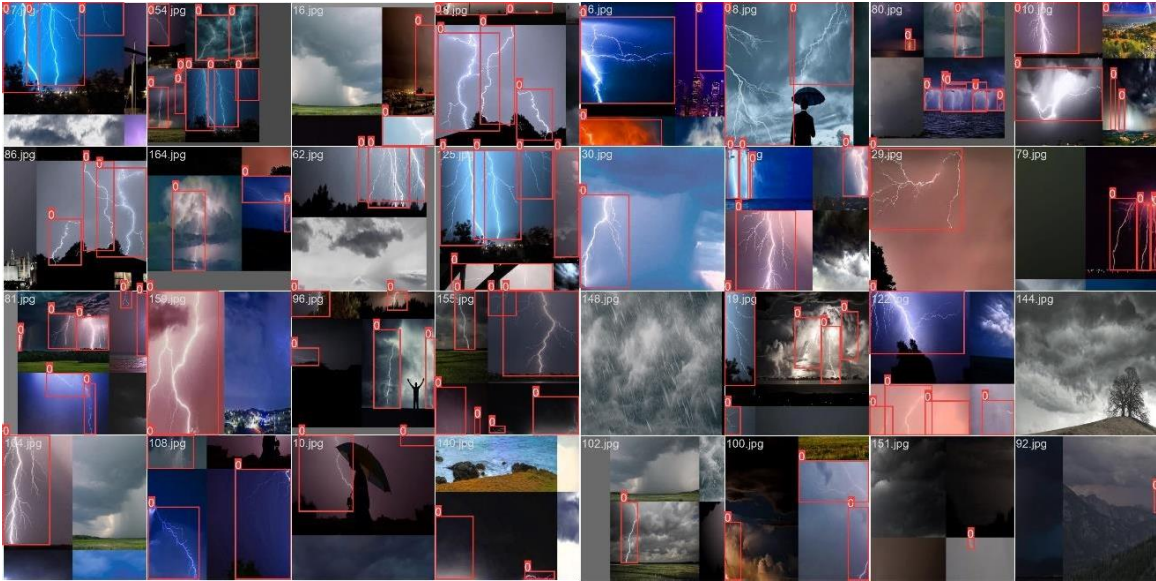
### 6.1. 1<sup>ο</sup> πειραματικό μοντέλο

#### 6.1.1. YOLOv5s.pt

Με το YOLOv5s.pt γίνεται χρήση του προεκπαιδευμένου μοντέλου YOLOv5 στην έκδοση small. Ακολουθεί η ανάλυση και η απόδοση του μοντέλου.

#### 6.1.1.1. Train batches

Το μοντέλο εκπαιδεύτηκε σύμφωνα με τις παραπάνω παραμέτρους που αναφέρθηκαν. Στο Σχήμα 37 παρατηρούνται τα train batches, παρέχοντας μια αναλυτική εικόνα της διαδικασίας εκπαίδευσης.



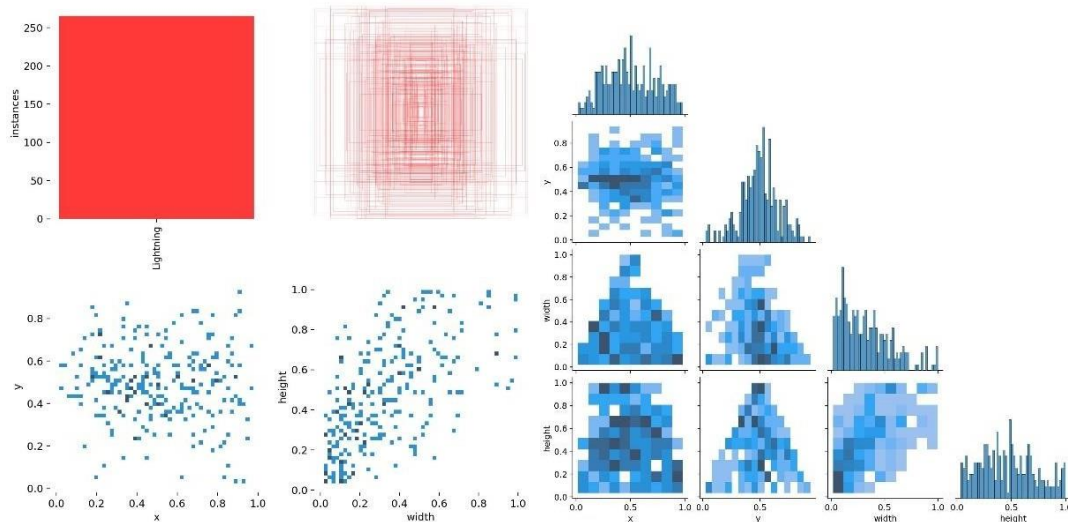
Σχήμα 37: Train batch 0 και 1 για το yolov5s.pt

Το train batch αναφέρεται σε ένα κομμάτι του συνόλου των δεδομένων (batch), το οποίο χρησιμοποιείται στην εκπαίδευση του μοντέλου. Συγκεκριμένα, το σύνολο των δεδομένων εκπαίδευσης χωρίζεται σε μικρότερα κομμάτια (batches), στο οποίο κάθε κομμάτι περιέχει ένα σύνολο παραδειγμάτων. Τα δεδομένα εκπαίδευσης χωρίζονται σε μικρότερα κομμάτια, ώστε να μειωθεί ο υπολογιστικός φόρτος και να επιταχυνθεί η διαδικασία εκπαίδευσης. Όσον αφορά την εκπαίδευση, το μοντέλο εκπαιδεύεται σε κάθε batch ξεχωριστά, ενώ μετά την εκπαίδευσή του σε αυτό, υπολογίζεται το gradient και το μοντέλο ενημερώνεται. Η διαδικασία αυτή επαναλαμβάνεται για όλα τα batches, έως ότου ολοκληρωθεί μια εποχή (epoch).

#### 6.1.1.2. Labels

Οι παρακάτω εικόνες απεικονίζουν τις στατιστικές αναλύσεις των δεδομένων των bounding boxes στο dataset για την ανίχνευση κεραυνών. Στην πρώτη εικόνα (Σχήμα 38), παρατηρείται

η κατανομή των instances ανά κατηγορία (με κόκκινο χρώμα) και η συγκέντρωση των bounding boxes στο χώρο (πάνω δεξιά). Συγκεκριμένα, τα περισσότερα bounding boxes συγκεντρώνονται στο κέντρο, το οποίο υποδηλώνει πως τα αντικείμενα εμφανίζονται κυρίως στη μέση των εικόνων.

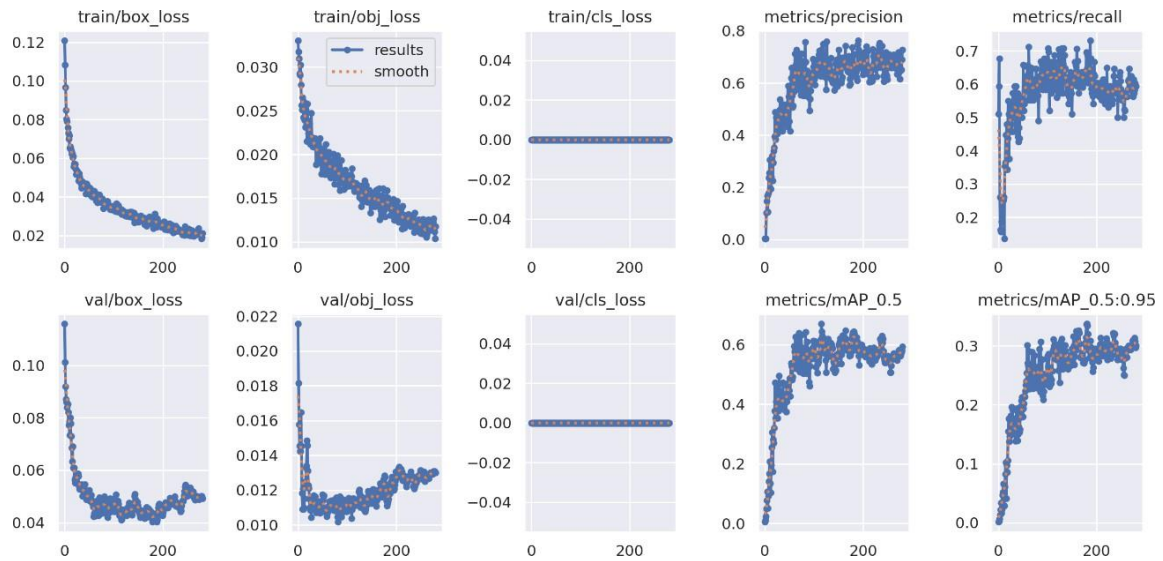


Σχήμα 38: Labels του yolov5s.pt

Στα κάτω διαγράμματα της πρώτης εικόνας, εμφανίζονται οι κατανομές θέσης ( $x$ ,  $y$ ) και διαστάσεων ( $width$ ,  $height$ ) των bounding boxes. Οι συντεταγμένες  $x$  και  $y$  είναι αρκετά διάσπαρτες αλλά όχι εντελώς ομοιόμορφες, ενώ το πλάτος και το ύψος των bounding boxes είναι συνήθως μικρά. Στη δεύτερη εικόνα (Σχήμα 38), απεικονίζεται ένα pair plot με πυκνογράμματα και ιστογράμματα για τις μεταβλητές  $x$ ,  $y$ ,  $width$  και  $height$ , για την οπτικοποίηση των συσχετίσεων μεταξύ τους. Το pair plot δείχνει πως υπάρχει συσχέτιση μεταξύ πλάτους και ύψους, διατηρώντας αναλογίες στα αντικείμενα.

### 6.1.1.3. Results

Τα αποτελέσματα της εκπαίδευσης αναλύονται και παρουσιάζονται παρακάτω:



Σχήμα 39: Τα αποτελέσματα της εκπαίδευσης του yolov5s.pt

**train/box\_loss:** Μειώνεται σταδιακά, υποδεικνύοντας πως το μοντέλο μαθαίνει να εντοπίζει καλύτερα τους κεραυνούς.

**train/obj\_loss:** Παρουσιάζει πτώση, το οποίο σημαίνει πως το μοντέλο αναγνωρίζει καλύτερα την ύπαρξη κεραυνών σε κάθε εικόνα.

**train/cls\_loss:** Είναι μηδενική, καθώς το μοντέλο δεν εκπαιδεύεται για κατηγοριοποίηση αντικειμένων, αφού υπάρχει μία κατηγορία, αυτή του κεραυνού.

**val/box\_loss:** Μειώνεται, όμως παρουσιάζει μικρές διακυμάνσεις, γεγονός το οποίο δείχνει ότι η γενίκευση δεν είναι απόλυτα ακριβής.

**val/obj\_loss:** Μειώνεται, αλλά παρουσιάζει ανοδικές τάσεις προς το τέλος, το οποίο μπορεί να υποδηλώνει μικρή υπερεκπαίδευση, όμως με το EarlyStopping που εφαρμόστηκε, αυτή εξαλείφθηκε όσο το δυνατόν περισσότερο.

**val/cls\_loss:** Μηδενική, όπως και στην εκπαίδευση.

**metrics/precision:** Αυξάνεται και σταθεροποιείται, το οποίο δείχνει βελτίωση.

**metrics/recall:** Βρίσκεται περίπου στο 0.7, το οποίο σημαίνει ότι το μοντέλο αναγνωρίζει τους κεραυνούς.

**metrics/mAP\_0.5:** Αυξάνεται και πλησιάζει περίπου το 0.65, παρουσιάζοντας τη συνολική επίδοση του μοντέλου.

**metrics/mAP\_0.5:0.95:** Αυξάνεται αλλά παραμένει σχετικά χαμηλό, το οποίο όμως αναμενόμενο λόγω αυστηρότερων κριτηρίων της μετρικής αυτής.

Στο τελικό στάδιο της διαδικασίας εκπαίδευσης, πραγματοποιείται η επικύρωση των αποτελεσμάτων, προκειμένου να αξιολογηθεί η απόδοση του μοντέλου. Στα Σχήματα 40-43, παρουσιάζονται οι τιμές των validation sets σε συνδυασμό με τις προβλεπόμενες τιμές που υπολογίστηκαν από το μοντέλο. Αυτή η σύγκριση μεταξύ των πραγματικών και των εκτιμώμενων αποτελεσμάτων προσφέρει μια σαφή εικόνα της ακρίβειας του μοντέλου και της δυνατότητάς του να προβλέπει σωστά τα δεδομένα.

#### 6.1.1.4. *Validation*

Το val αναφέρεται στο σετ επικύρωσης (validation set) κατά τη διάρκεια της εκπαίδευσης του μοντέλου. Αυτό το σετ δεδομένων χρησιμοποιείται για να αξιολογηθεί η απόδοση του μοντέλου μετά από κάθε εποχή εκπαίδευσης. Στόχος της επικύρωσης αποτελεί η αξιολόγηση του μοντέλου στη γενίκευση σε άγνωστα δεδομένα, στα οποία δεν έχει εκπαιδευτεί. Τα δεδομένα επικύρωσης δεν χρησιμοποιούνται για την εκπαίδευση του μοντέλου, αλλά μόνο για την αξιολόγηση της απόδοσης του.

Το val\_pred αναφέρεται στις προβλέψεις του μοντέλου κατά τη διάρκεια της αξιολόγησης, σύμφωνα με το σετ επικύρωσης, επομένως οι προβλέψεις αυτές αναφέρονται στα αποτελέσματα του μοντέλου για τα δεδομένα επικύρωσης. Οι προβλέψεις του μοντέλου συγκρίνονται με τα πραγματικά αποτελέσματα του σετ επικύρωσης, ώστε να υπολογιστεί η συνολική του απόδοση.





Σχήμα 40: Σειρά επικύρωσης 0 για το yolov5s.pt

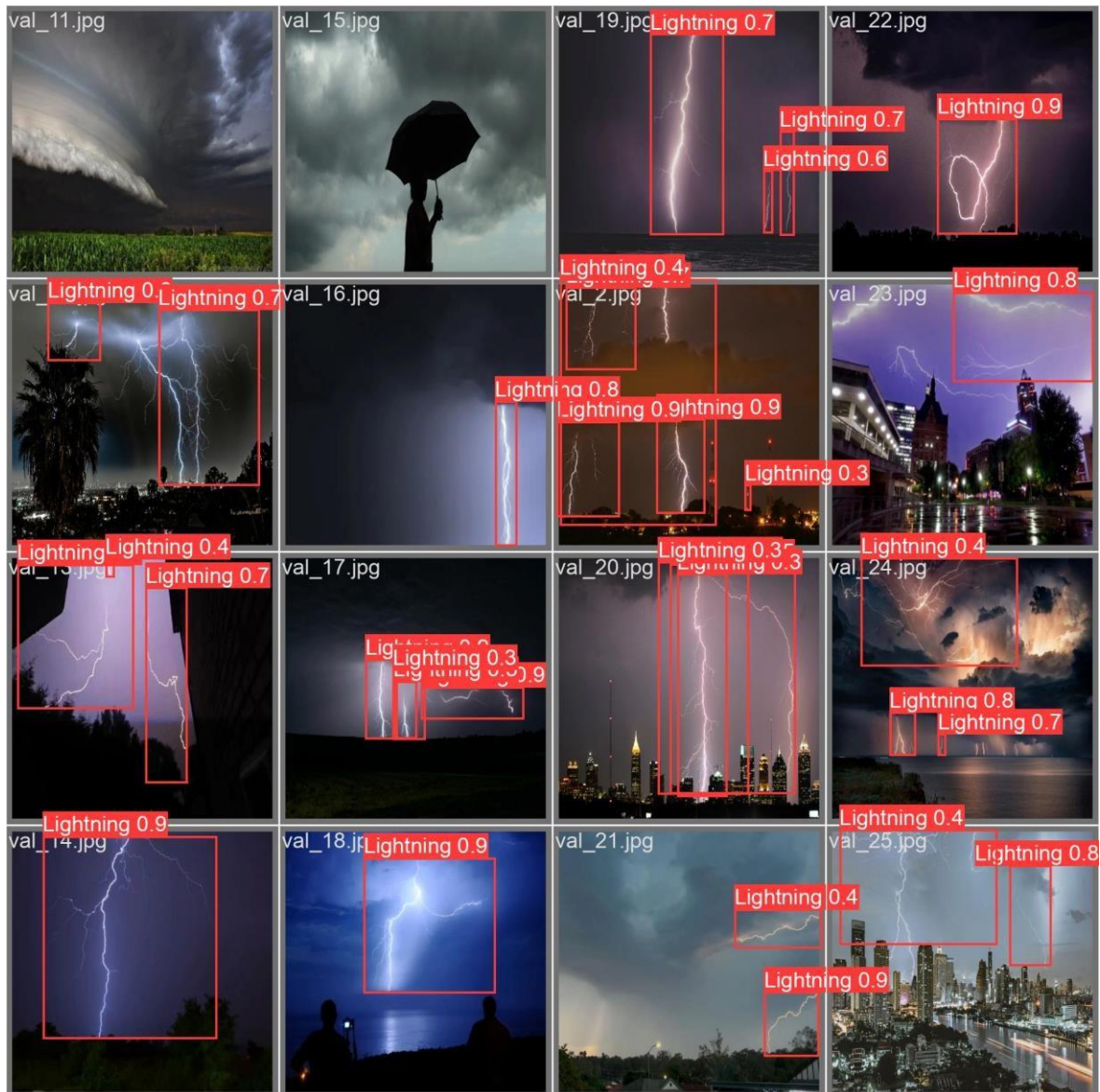


Σχήμα 41: Προβλέψεις του σετ επικύρωσης 0 για το yolov5s.pt





Σχήμα 42: Σετ επικύρωσης 1 για το yolov5s.pt



Σχήμα 43: Προβλέψεις του σετ επικύρωσης 1 για το yolov5s.pt

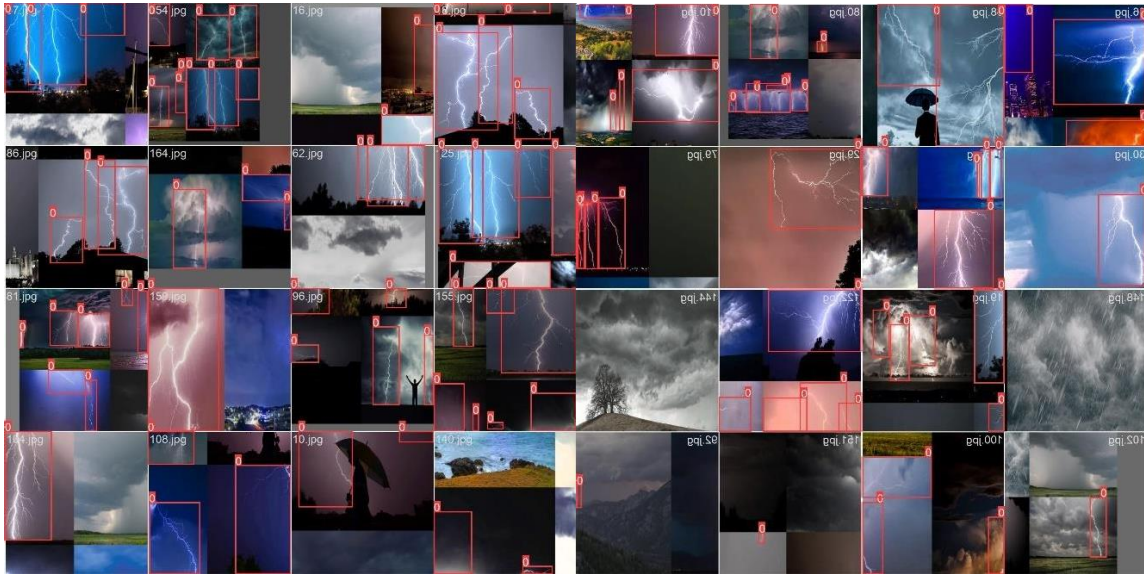
Παρατηρείται πως το μοντέλο δεν εντοπίζει ορισμένα bounding boxes, το οποίο προκύπτει και από τα αποτελέσματα και υποδεικνύει πως η ακρίβειά του δεν φτάνει το 100%. Βέβαια, η απόδοσή του βρίσκεται σε ένα πολύ καλό επίπεδο, προσφέροντας αξιόπιστα αποτελέσματα για την πλειονότητα των περιπτώσεων.



### 6.1.2. YOLOv5m.pt

#### 6.1.2.1. Train batches

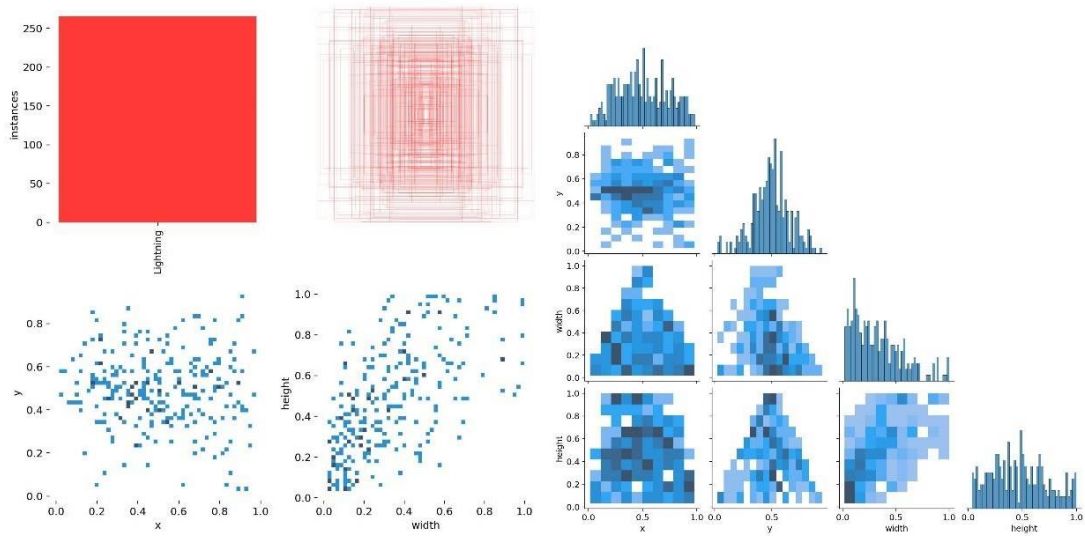
Παρακάτω εμφανίζονται τα training batch 0 και 1 για το μοντέλο YOLOv5m.pt.



Σχήμα 44: Train batch 0 και 1 για το yolov5m.pt

#### 6.1.2.2. Labels

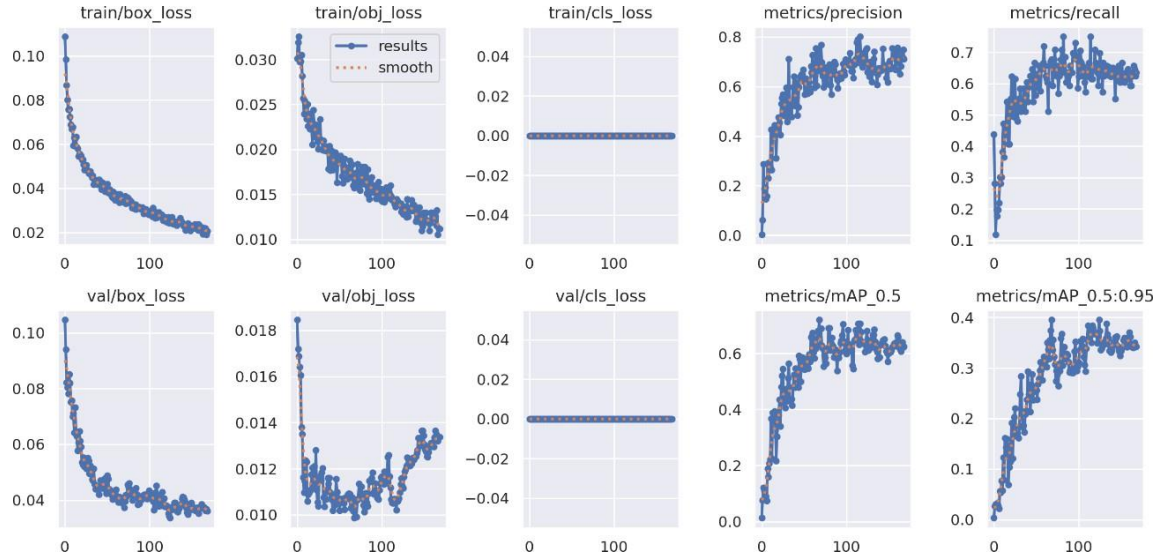
Αντίστοιχα, τα παρακάτω παρατηρούνται για το μοντέλο yolov5m.pt.



Σχήμα 45: Labels του yolov5m.pt

### 6.1.2.3. Results

Παρατηρείται βελτίωση στις μετρικές, σε σχέση με το 1<sup>ο</sup> μοντέλο. Στη δεύτερη εικόνα, το μοντέλο εκπαιδεύτηκε για λιγότερες εποχές αλλά φαίνεται να έχει καλύτερη απόδοση από το πρώτο. Το val/box\_loss και το val/obj\_loss σταθεροποιούνται πιο γρήγορα, το οποίο δηλώνει πως στο δεύτερο μοντέλο παρατηρείται καλύτερη γενίκευση.



Σχήμα 46: Τα αποτελέσματα της εκπαίδευσης του yolov5m.pt

Το precision και το recall ακολουθούν παρόμοια εμφάνιση και στις δύο περιπτώσεις, ωστόσο στη δεύτερη περίπτωση, η σύγκλιση φαίνεται να είναι ελαφρώς ταχύτερη. Το mAP@0.5 και mAP@0.5:0.95 αυξάνονται και στις δύο περιπτώσεις, όμως στο δεύτερο μοντέλο το τελικό mAP είναι ελαφρώς υψηλότερο.

#### 6.1.2.4. Validation



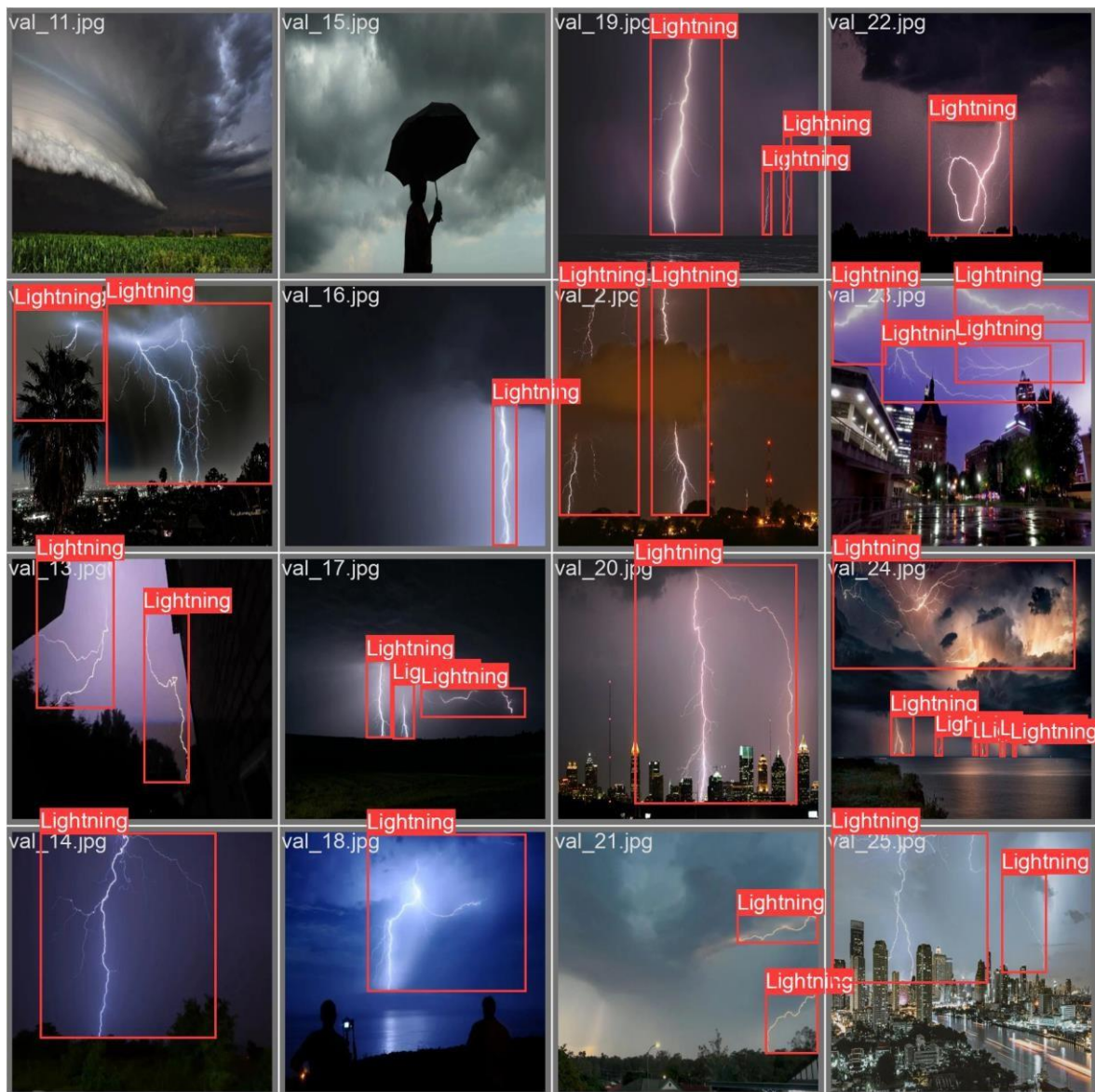
Σχήμα 47: Σειτ επικόρωσης 0 για το yolov5m.pt



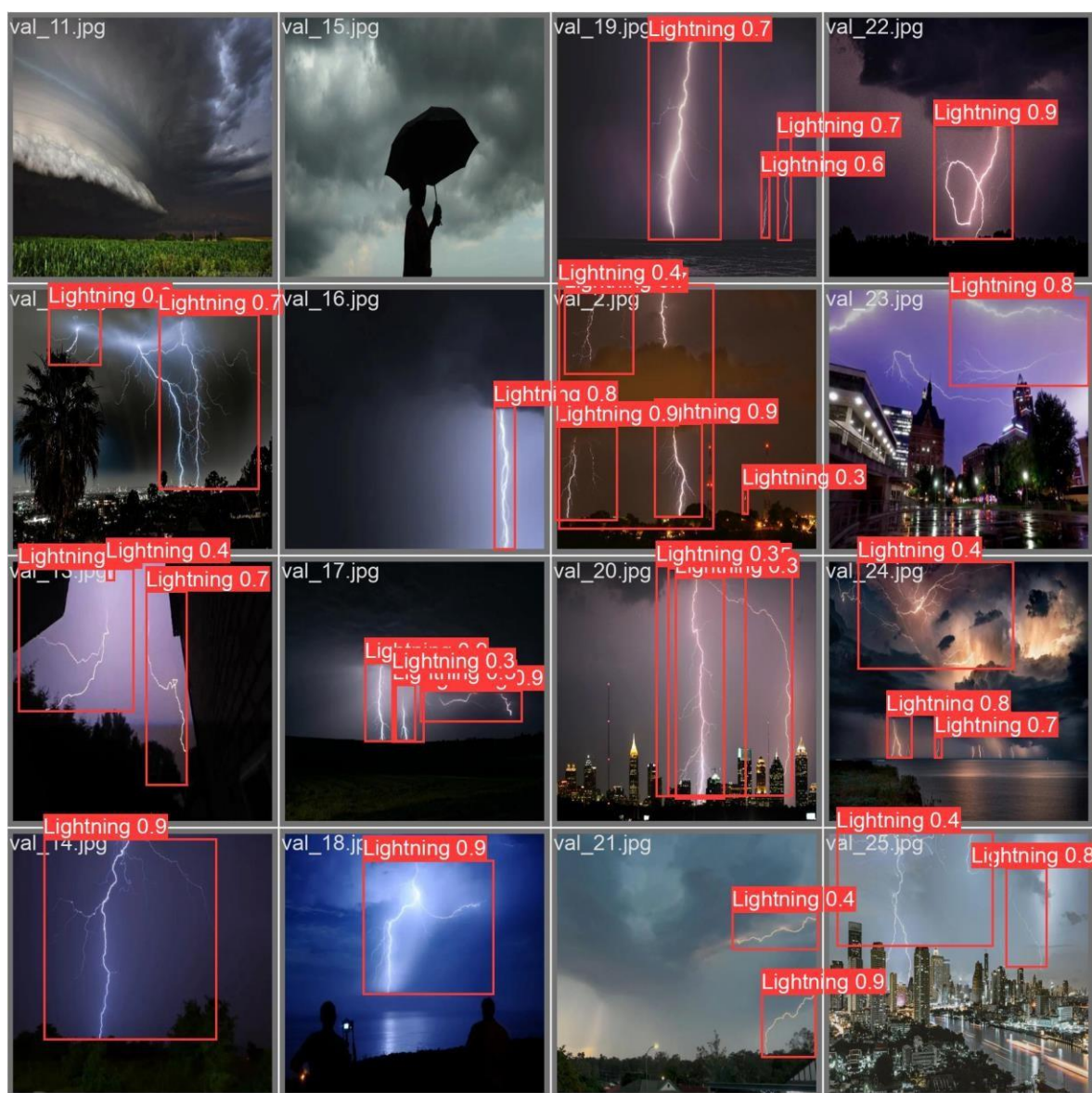


Σχήμα 48: Προβλέψεις του σετ επικύρωσης 0 για το yolov5m.pt





Σχήμα 49: Σετ επικύρωσης 1 για το yolov5m.pt

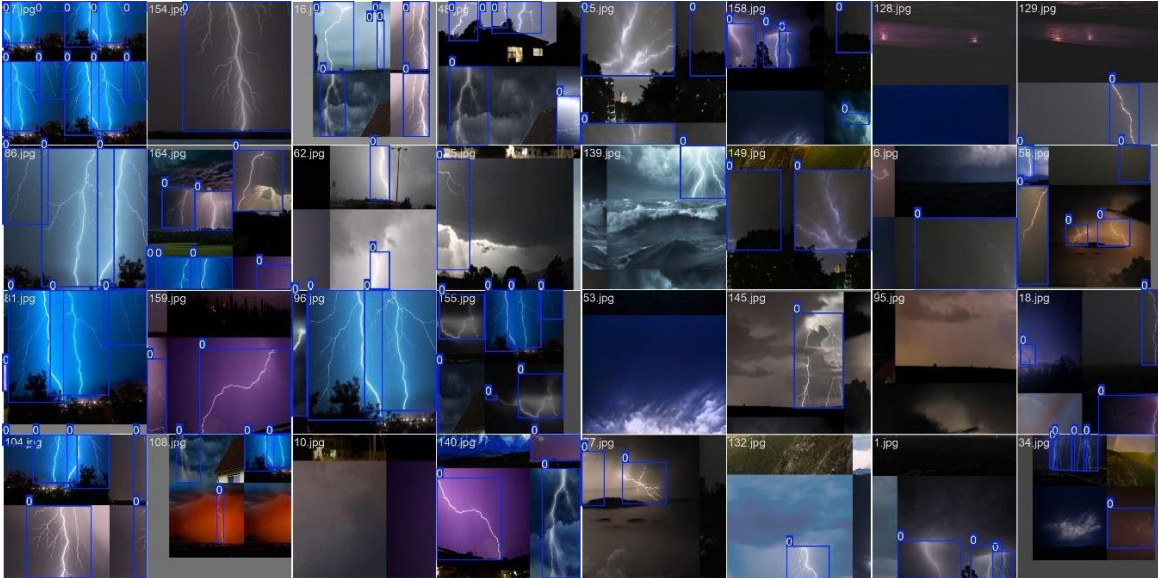


Σχήμα 50: Προβλέψεις του σετ επικύρωσης 1 για το yolov5m.pt

## 6.2. Το 2<sup>ο</sup> πειραματικό μοντέλο

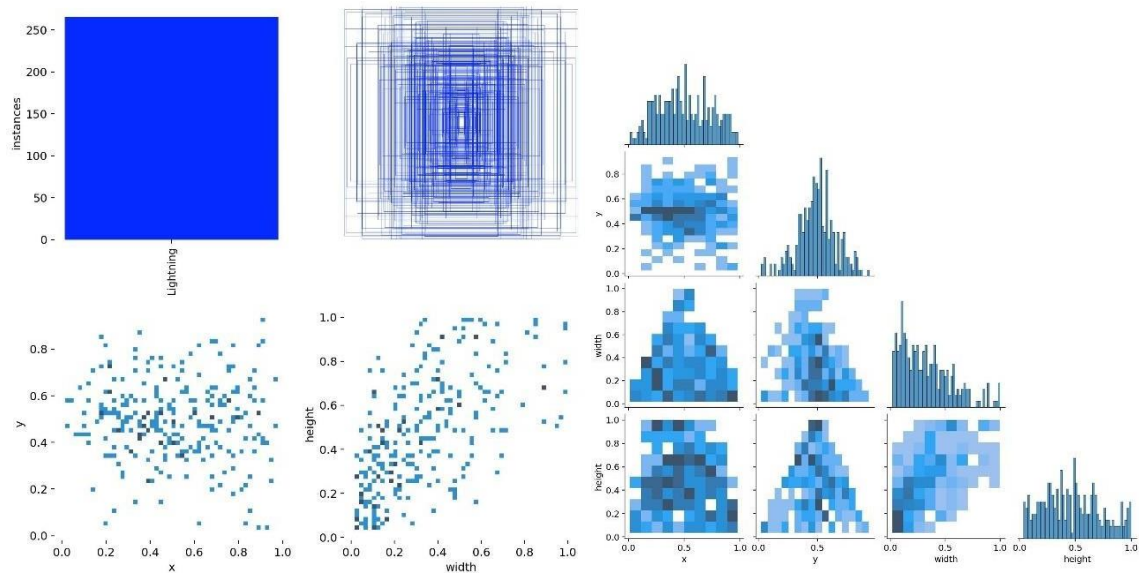
### 6.2.1. yolov8s.pt

#### 6.2.1.1. Train batches



Σχήμα 51: Train batch 0 και 1 για το yolov8s.pt

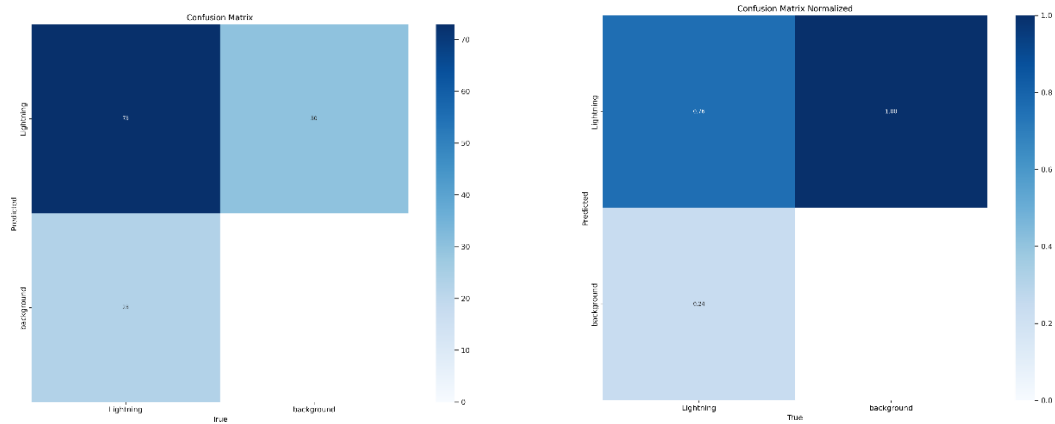
#### 6.2.1.2. Labels



Σχήμα 52: Labels του yolov8s.pt



### 6.2.1.3. Confusion matrices

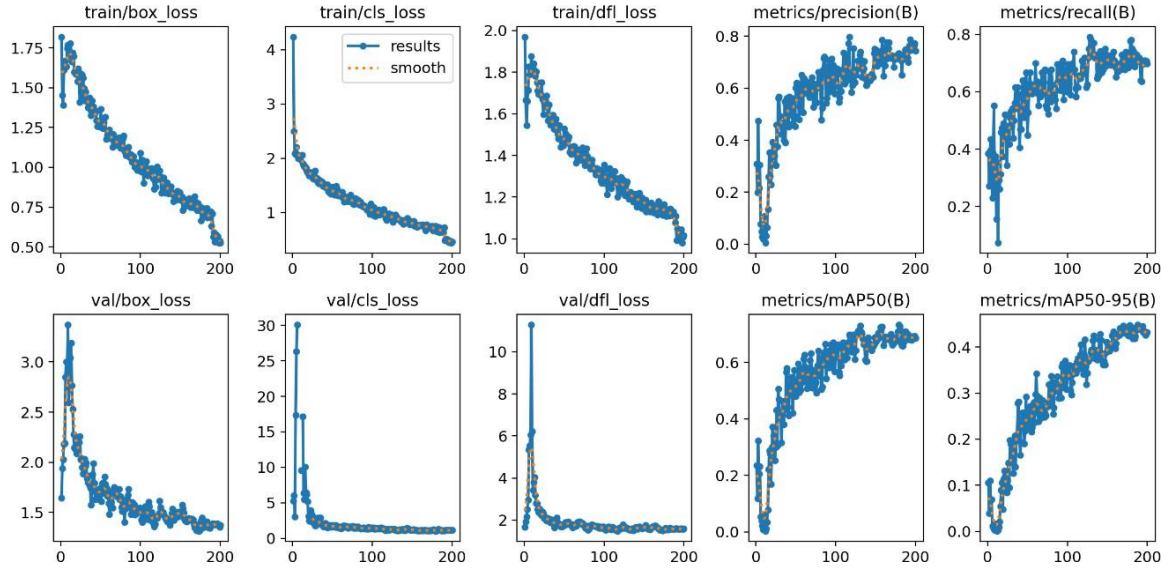


Σχήμα 53: Confusion matrices του yolov8s.pt

Οι παραπάνω πίνακες, αναφέρονται στους πίνακες σύγχυσης (confusion matrices), οι οποίοι αξιολογούν την απόδοση του μοντέλου στην ταξινόμηση μεταξύ των κατηγοριών Lightning και background. Ο πρώτος πίνακας είναι ο βασικός πίνακας σύγχυσης, ο οποίος δείχνει τον αριθμό των σωστών και λανθασμένων προβλέψεων. Ο δεύτερος πίνακας είναι ο κανονικοποιημένος πίνακας σύγχυσης, στον οποίο οι τιμές είναι σε ποσοστά από 0 έως 1.

- 76% από τις πραγματικές τιμές Lightning ανιχνεύτηκαν σωστά.
- 100% των πραγματικών τιμών background ανιχνεύτηκαν σωστά.
- 24% των πραγματικών τιμών Lightning δεν αναγνωρίστηκαν σωστά (False Negatives).

#### 6.2.1.4. Results



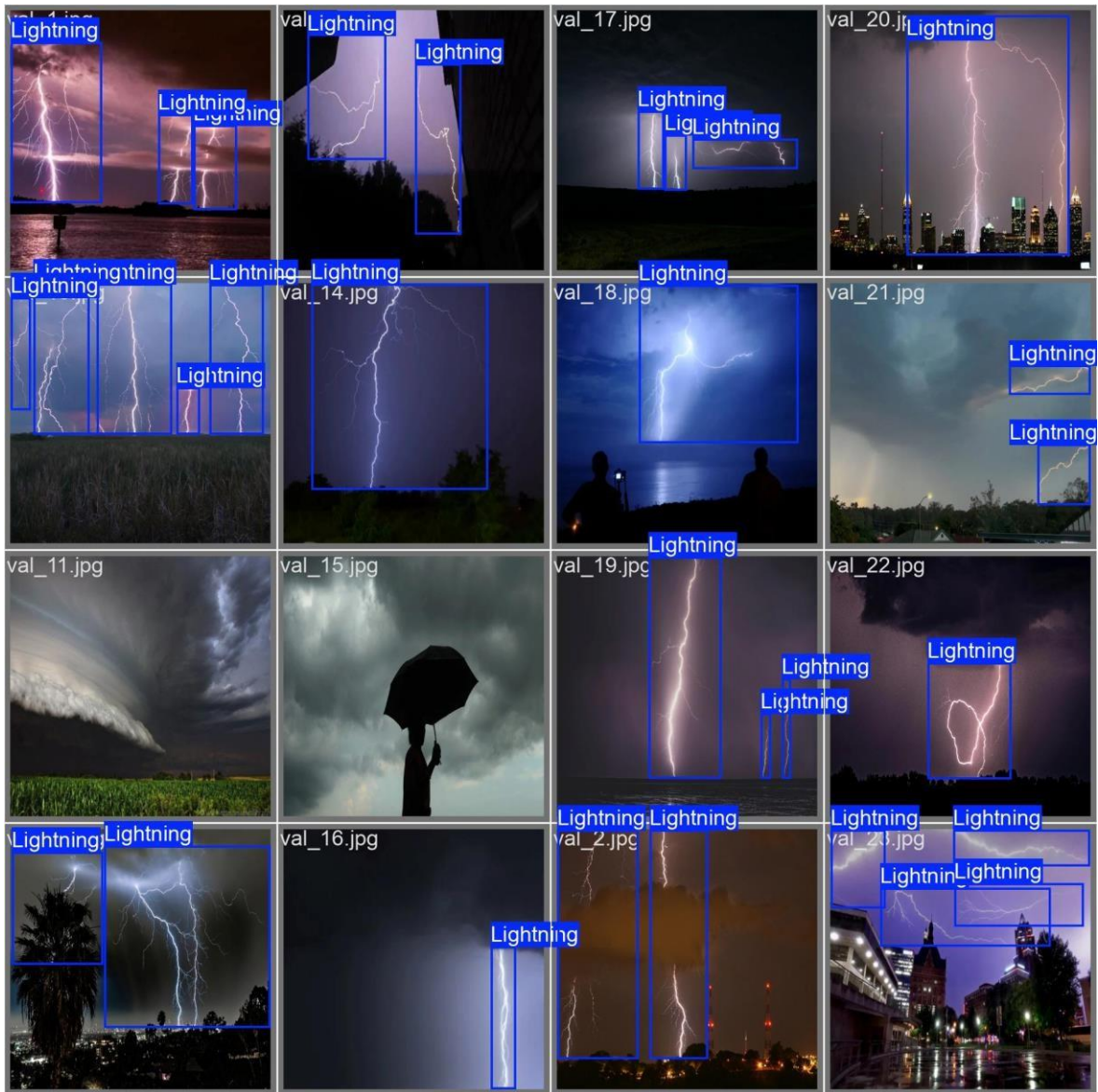
Σχήμα 54: Τα αποτελέσματα της εκπαίδευσης του yolov8s.pt

metrics/precision(B): Παρουσιάζεται κοντά στο 0.8, υποδεικνύοντας πως το μοντέλο γίνεται πιο ακριβές στις προβλέψεις του με την εκπαίδευση.

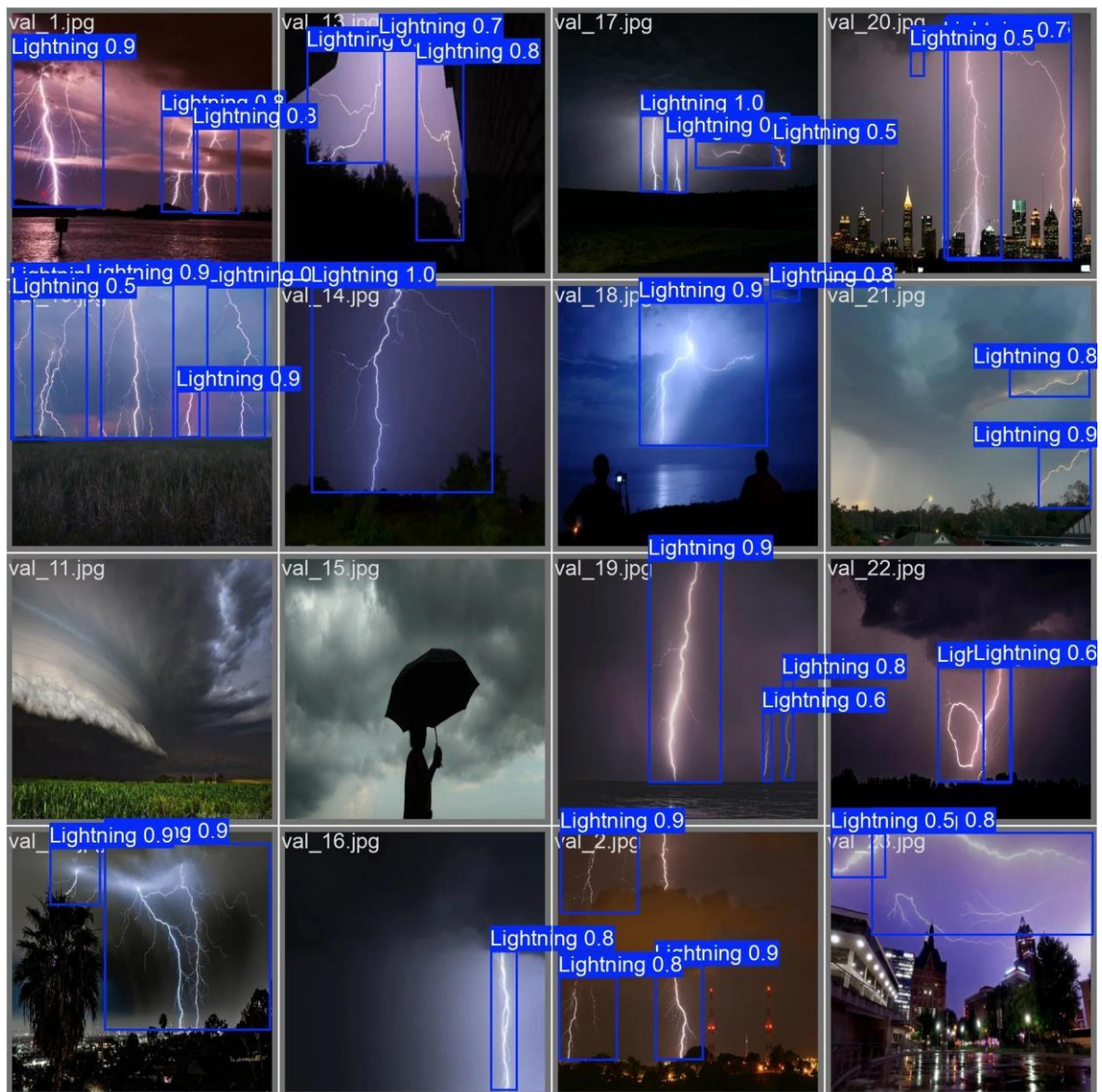
metrics/recall(B): Αυξάνεται σταθερά και φτάνει σε τιμή κοντά στο 0.8, επομένως το μοντέλο ανιχνεύει περισσότερα αντικείμενα με την πάροδο των εποχών.

metrics/mAP50(B) και mAP50-95(B): Οι τιμές αυξάνονται σταδιακά, γεγονός το οποίο σημαίνει ότι η συνολική ακρίβεια του μοντέλου βελτιώνεται με τον χρόνο. Η πρώτη μετρική εντοπίζεται κοντά στο 0.7 (συγκεκριμένα στο 0.709), ενώ η δεύτερη κοντά στο 4.5.

#### 6.2.1.5. Validation



Σχήμα 55: Σετ επικύρωσης 0 για το yolov8s.pt



Σχήμα 56: Προβλέψεις του σετ επικύρωσης 0 για το yolov8s.pt





Σχήμα 57: Σειρά επικύρωσης 1 για το yolov8s.pt

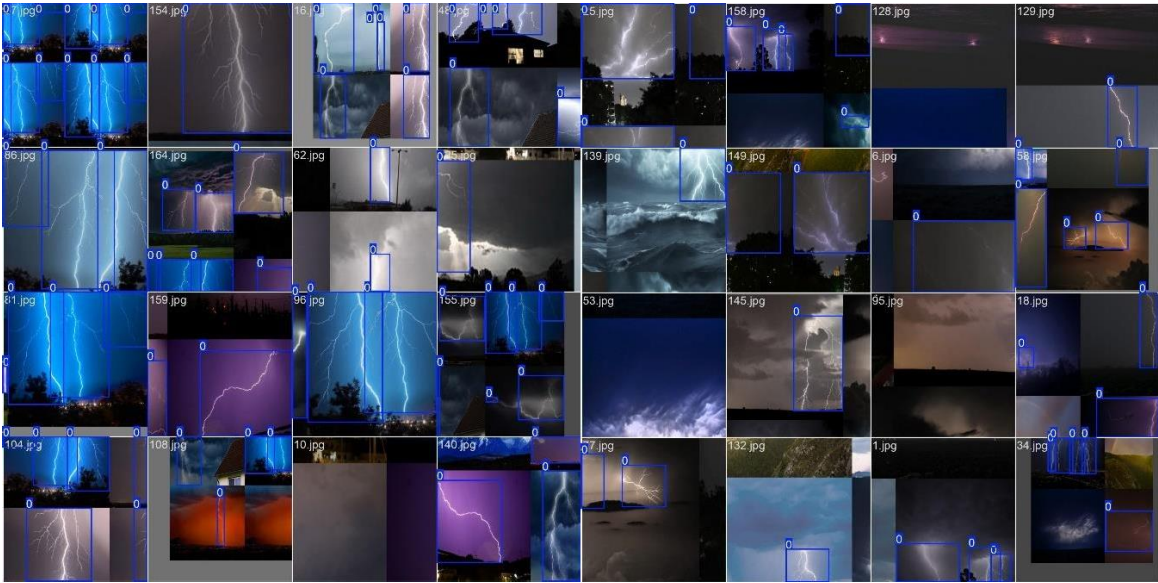




Σχήμα 58: Προβλέψεις του σετ επικύρωσης 1 για το yolov8s.pt

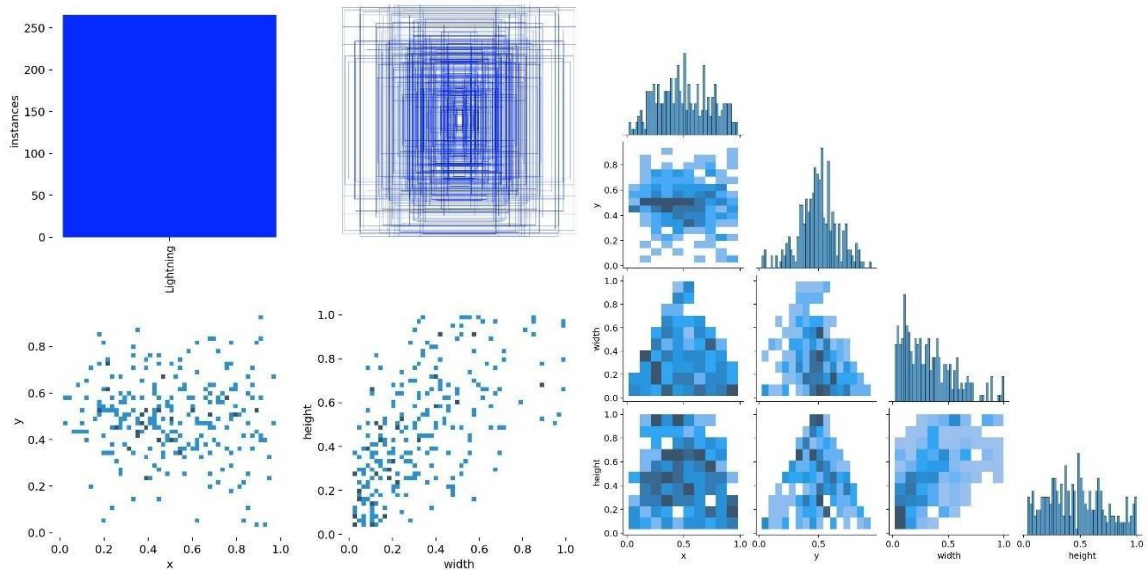
## 6.2.2. Yolov8m.pt

### 6.2.2.1. Train batches



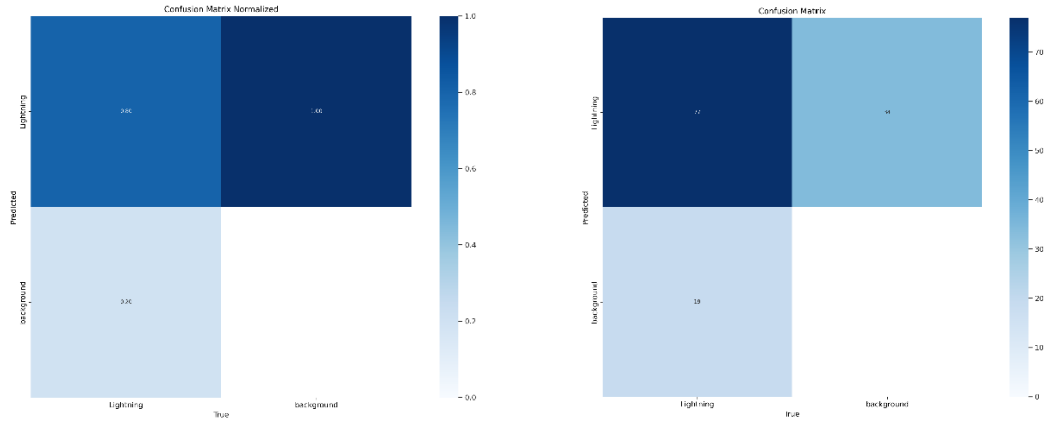
Σχήμα 59: Train batch 0 και 1 για το yolov8m.pt

### 6.2.2.2. Labels



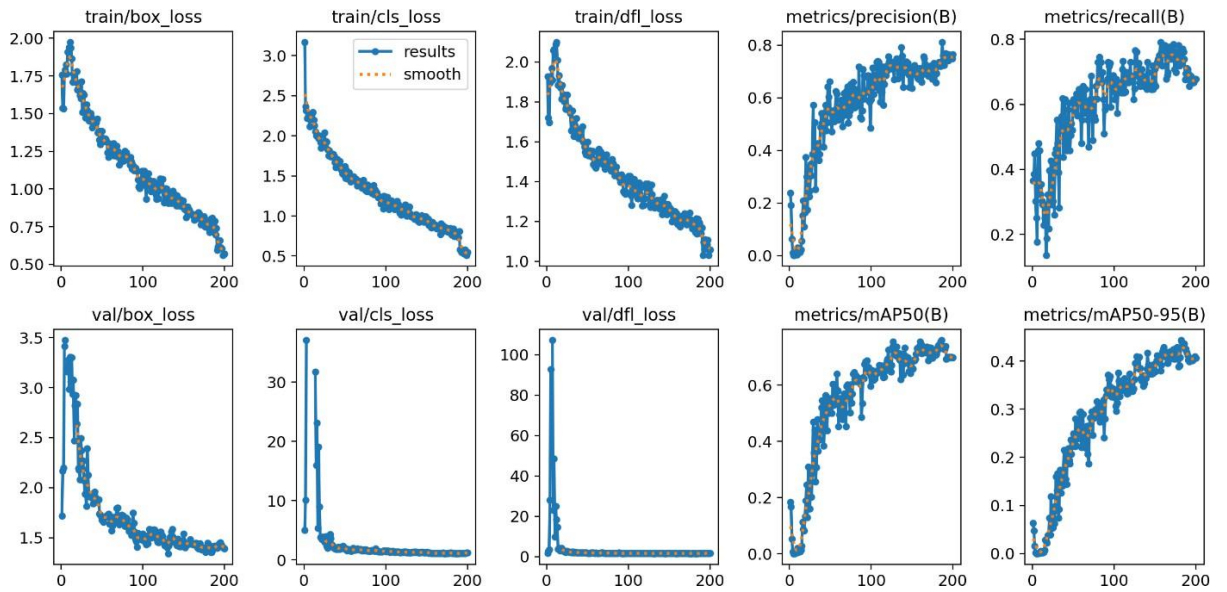
Σχήμα 60: Labels του yolov8m.pt

### 6.2.2.3. Confusion matrices



Σχήμα 61: Confusion matrices του yolov8m.pt

### 6.2.2.4. Results



Σχήμα 62: Τα αποτελέσματα της εκπαίδευσης του yolov8m.pt

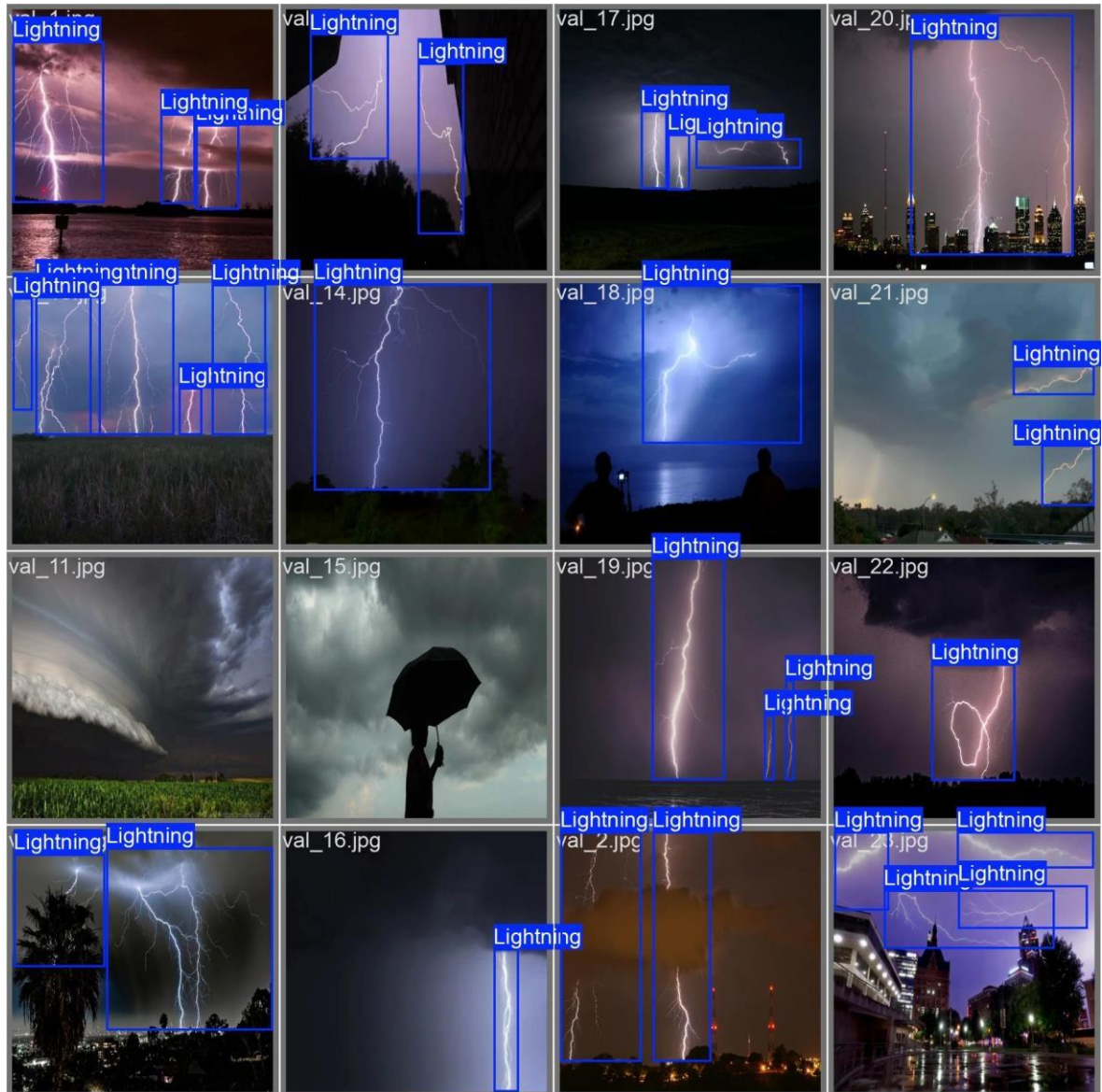
Η precision, η recall, η mAP50 και η mAP50-95 παρουσιάζουν παρόμοιες τιμές με το προηγούμενο μοντέλο, με σταθερή βελτίωση. Πιο συγκεκριμένα, η precision και η recall πλησιάζουν τις τιμές κοντά στο 0.8, η mAP50, όπως και στο προηγούμενο μοντέλο βρίσκεται περίπου στο 0.7, ενώ η mAP50-95, η αυστηρότερη μετρική, γύρω στο 0.45.

Γενικά, ανάμεσα στα δύο μοντέλα δεν παρουσιάζονται σημαντικές διαφορές σε απώλειες και μετρικές. Κύρια διαφορά μεταξύ των δύο μοντέλων, όμως, αποτελεί ο χρόνος

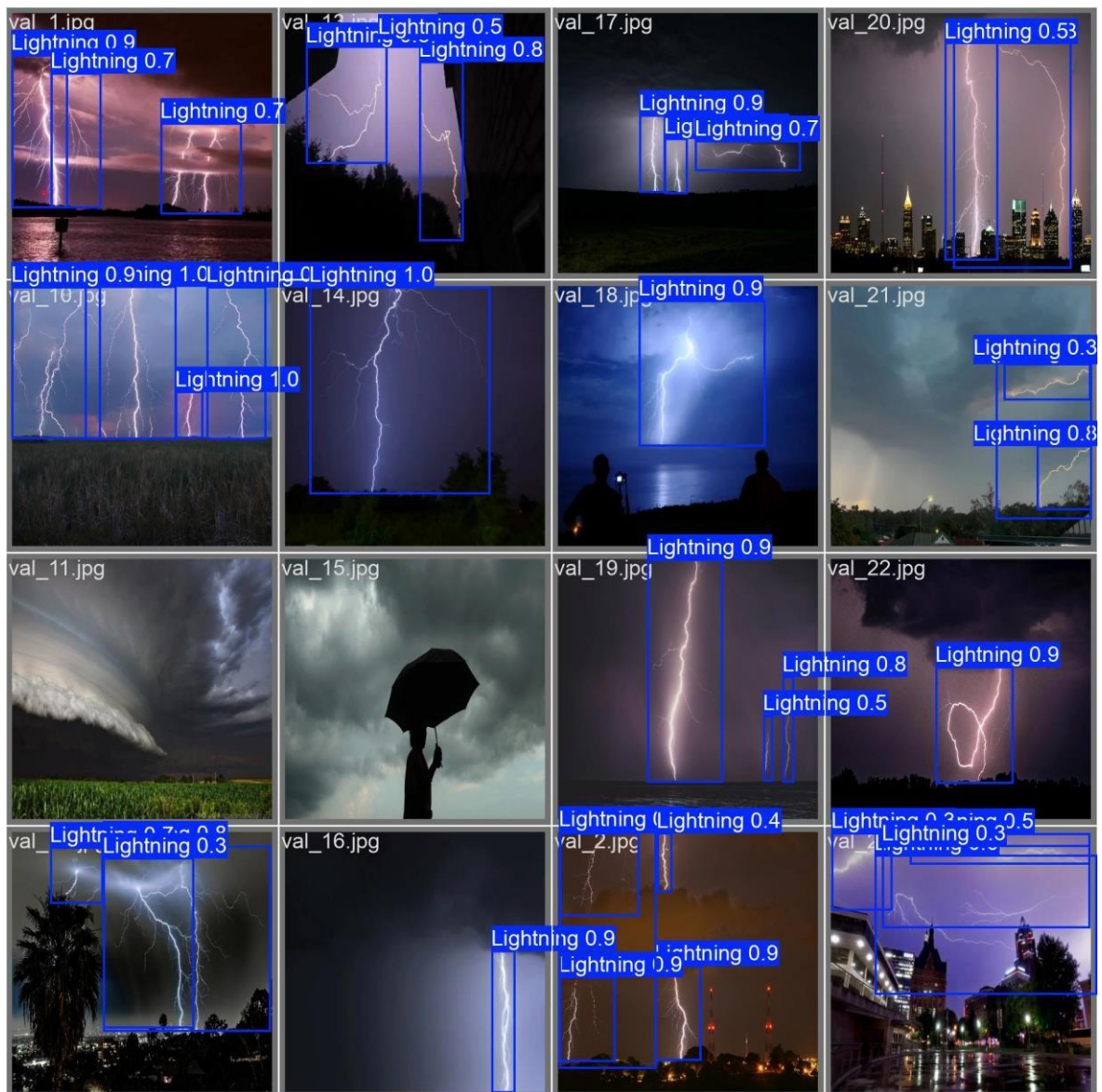


της ολοκλήρωσης της εκπαίδευσης. Στο πρώτο μοντέλο η εκπαίδευση των εποχών ολοκληρώθηκε σε 0.266 ώρες, ενώ στο δεύτερο σε 0.465 ώρες.

#### 6.2.2.5. Validation



Σχήμα 63: Σετ επικύρωσης 0 για το yolov8m.pt



Σχήμα 64: Προβλέψεις του σετ επικύρωσης 0 για το yolov8m.pt





Σχήμα 65: Σειρά επικύρωσης 1 για το yolov8m.pt



Σχήμα 66: Προβλέψεις του σετ επικύρωσης 1 για το yolov8m.pt

## 7. Συμπεράσματα

Αναλύοντας όλες τις παραπάνω πληροφορίες και για τα δύο πειραματικά μοντέλα στις δύο τους εκδόσεις, το YOLOv5 και το YOLOv8, οι οποίες εξήχθησαν από τα διαγράμματα και τις εικόνες, παρατηρείται πως και τα δύο μοντέλα, με τις αντίστοιχες εκδόσεις small και medium, δεν εμφανίζουν σημαντικές και με μεγάλες διακυμάνσεις διαφορές στις μετρικές. Παρατηρήθηκε πως, η μεσαία έκδοση του μοντέλου, δεν παρείχε αξιόλογες βελτιώσεις στην απόδοση, σε σχέση με τη μικρότερη έκδοση. Κύρια διαφορά μεταξύ των δύο μοντέλων, όμως, αποτελεί ο χρόνος της ολοκλήρωσης της εκπαίδευσης. Στο πρώτο μοντέλο η εκπαίδευση των εποχών ολοκληρώθηκε σε 0.266 ώρες, ενώ στο δεύτερο σε 0.465 ώρες.

Βέβαια, αναλόγως την έκδοση υπάρχει και η αντίστοιχη διαφορά χρόνου στην εκπαίδευση που είναι αναγκαίο να ληφθεί υπόψιν. Είναι προφανές πως όσο μικρότερος είναι ο χρόνος εκπαίδευσης, τόσο πιο ωφέλιμο είναι για το μοντέλο, καθώς μειώνει τη κατανάλωση υπολογιστικών πόρων. Επομένως, η μείωση του χρόνου εκπαίδευσης συμβάλλει στη μείωση του ενεργειακού αποτυπώματος, καθώς οι χρονοβόρες εκπαιδεύσεις καταναλώνουν σημαντική ποσότητα ηλεκτρικής ενέργειας. Ακόμη, είναι απαραίτητο ο πειραματισμός να γίνεται όσο το δυνατόν συντομότερα, καθώς μικρότεροι χρόνοι εκπαίδευσης επιτρέπουν γρήγορες επαναλήψεις και δοκιμές, ενώ σε περιπτώσεις που απαιτείται εκπαίδευση σε πραγματικό χρόνο, οι μικρότεροι χρόνοι εκπαίδευσης είναι κρίσιμοι.

Παρά το γεγονός ότι η μετρική  $mAP@50:95$  παρουσιάζει χαμηλές τιμές, αυτό δεν υποδηλώνει πως είναι ανεπαρκής η απόδοση του μοντέλου, αλλά βασίζεται στην αυστηρότητα της συγκεκριμένης μετρικής. Επιπλέον, η φύση του φαινομένου του κεραυνού προσθέτει προκλήσεις στην ανίχνευση, καθώς εμφανίζεται στιγμιαία, με έντονες μεταβολές στη φωτεινότητα, το περιβάλλον ανίχνευσης ποικίλει, ενώ το σημαντικότερο για την ανάπτυξη μοντέλου ανίχνευσής τους είναι πως η μορφολογία τους δεν ακολουθεί σταθερά πρότυπα, γεγονός που δυσκολεύει την ακριβή περιγραφή τους μέσω προκαθορισμένων anchor boxes. Ωστόσο, οι υπόλοιπες μετρικές απόδοσης, η precision, η recall και η  $mAP@50$ , εμφανίζουν τιμές μεταξύ 0.7 και 0.8, το οποίο υποδεικνύει πως το μοντέλο επιτυγχάνει υψηλή ανίχνευση όταν χρησιμοποιείται ένα πιο χαμηλό κατώφλι IoU.

Ένα σημαντικός παράγοντας, στον οποίο μπορεί να οφείλεται μία χαμηλότερη επίδοση στις μετρικές, ιδιαίτερα στη μετρική  $mAP@50:95$ , είναι η φύση του ίδιου του



κεραυνού και η μέθοδος με την οποία σημειώθηκαν τα bounding boxes στα δεδομένα εκπαίδευσης. Είναι πιθανό το μοντέλο να ανιχνεύει σωστά την παρουσία των κεραυνών, αλλά τα παραγόμενα bounding boxes να διαφέρουν από αυτά που έχουν επισημανθεί χειροκίνητα. Αυτό μπορεί να οφείλεται στο γεγονός πως οι κεραυνοί έχουν ασαφή όρια και πολύπλοκες δομές, οπότε ο καθορισμός ενός ακριβούς bounding box μπορεί να είναι υποκειμενικός. Εάν το μοντέλο ανιχνεύει σωστά την παρουσία κεραυνών, αλλά οι παραγόμενες περιοχές δεν ευθυγραμμίζονται ακριβώς με τις ετικέτες, τότε το IoU θα είναι χαμηλότερο, οδηγώντας σε χαμηλότερες μετρικές.

Στην ανάλυση των αποτελεσμάτων στο validation set, φαίνεται πως όλοι οι κεραυνοί ανιχνεύθηκαν, με διαφορετικά όμως bounding boxes, το οποίο θα μπορούσε να επιβεβαιώνει ότι το μοντέλο αναγνωρίζει το αντικείμενο, αλλά οι διαφορές στις συντεταγμένες των bounding boxes επηρεάζουν τις μετρικές.

Τα νεότερα μοντέλα YOLOv10 και YOLOv12, αν και αποτελούν τις πιο πρόσφατες εκδόσεις της οικογένειας YOLO, δεν παρουσίασαν καλύτερη απόδοση στην ανίχνευση κεραυνών σε σύγκριση με τα YOLOv5 και YOLOv8, όταν πραγματοποιήθηκε εκπαίδευση του dataset σε αυτά. Το γεγονός αυτό, μπορεί να οφείλεται σε διάφορους παράγοντες που σχετίζονται τόσο με την αρχιτεκτονική των μοντέλων αυτών όσο και με τη φύση της ίδιας της ανίχνευσης κεραυνών.

Τα αποτελέσματα δείχνουν ότι, παρότι τα YOLOv10 και YOLOv12 ενσωματώνουν προηγμένες τεχνολογίες, τα YOLOv5 και YOLOv8 εξακολουθούν να υπερτερούν στην ανίχνευση κεραυνών, καθώς έχουν χρησιμοποιηθεί ευρέως και έχουν βελτιωθεί μέσα από διάφορες εφαρμογές. Καλύτερη επίδοση, συνολικά, σε χρόνο αλλά και απόδοση του μοντέλου, διαπιστώθηκε στο μοντέλο YOLOv8s, όπως προκύπτει από την ανάλυση των αποτελεσμάτων.

Το dataset που χρησιμοποιήθηκε αποτελείται από 230 εικόνες, αριθμός επαρκής για μία αρχική προσέγγιση στην ανίχνευση κεραυνών, αλλά με δυνατότητα βελτίωσης. Ως προοπτική για περαιτέρω έρευνα, προτείνεται η διεύρυνση του συνόλου δεδομένων με επιπλέον αντιπροσωπευτικές εικόνες κεραυνών, ώστε να ενισχυθεί η γενίκευση του μοντέλου, το οποίο μπορεί να συμβάλει στην βελτίωση της απόδοσης των νεότερων YOLO μοντέλων.

## BIBΛΙΟΓΡΑΦΙΑ

- 1) Anders A., (2003), Tracking Down the Origin of Arc Plasma Science I. Early Pulsed and Oscillating Discharges, IEEE Transactions on Plasma Science 31 (4): 1052–1059, doi:10.1109/TPS.2003.815476
- 2) Balkhanov V. K., Bashkuev Y., (2012), Fractal-geometry simulation of a lightning discharge, Technical Physics, 57(12), 1593–1597, <https://doi.org/10.1134/S106378421212002X>
- 3) Britannica, T. Editors of Encyclopaedia (2025), Fractal, Encyclopedia Britannica, <https://www.britannica.com/science/fractal>
- 4) Brune W. H., McFarland P. J., Bruning E., Waugh S., MacGorman D., Miller D. O., Jenkins J. M., Ren X., Mao J., Peischl J., (2021), Extreme oxidant amounts produced by lightning in storm clouds, Science, 372(6543), 711–715, <https://doi.org/10.1126/science.abg0492>
- 5) Bullis L., Pundure J., Jemeljanovs V., (2023)., Lightning protection of buildings: Its types, solutions, and choice options, State Fire and Rescue Service, Riga, Latvia; Riga Technical University, Riga, Latvia
- 6) Cooray V., (2014), An Introduction to Lightning, Springer Verlag, doi:10.1007/978-94-017-8938-7, ISBN 978-94-017-8937-0
- 7) Field P. R., Hand W. H., Cappelluti G., McMillan A., Foreman A., Stubbs D., Willows M., (2010), Hail Threat Standardisation, (PDF), European Aviation Safety Agency, Research Project EASA.2008/5
- 8) Fisher F. A., Plurner J. A., (1980), Lightning protection of aircraft (NASA Reference Publication 1008), National Aeronautics and Space Administration, Scientific and Technical Information Office
- 9) Fractal Foundation, (n.d.), Fractal Lightning, Retrieved from <https://fractalfoundation.org>
- 10) Giancoli D. C., (1998), Physics: principles with applications, 5th ed, Upper Saddle River, N.J., Prentice Hall
- 11) Gosline A., (2005), Thunderbolts from space, New Scientist, 186 (2498): 30–34
- 12) Haque K. N., (2023), What is Convolutional Neural Network — CNN (Deep Learning), LinkedIn, Retrieved from <https://www.linkedin.com>

- 13) Han S. H., Kim K.W., Kim S., Youn Y.C., (2018), Artificial Neural Network: Understanding the Basic Concepts without Mathematics, *Dementia Neurocognitive Disorders*, 17(3):83-89. doi: 10.12779/dnd.2018.17.3.83
- 14) Higashi Y., Yamamoto K., (2023), Study of Lightning Detection Method Using Acoustic Data, 12th Asia-Pacific International Conference on Lightning (APL), Langkawi, Malaysia, 2023, pp. 1-5, doi: 10.1109/APL57308.2023.10181611
- 15) Holle R. H., López, R. E., (1993), Overview of real-time lightning detection systems and their meteorological uses (NOAA Technical Memorandum ERL NSSL No. 102), National Severe Storms Laboratory; National Oceanic and Atmospheric Administration, Retrieved from <https://repository.library.noaa.gov/view/noaa/11301>
- 16) Ierides A., Suddarth S., Kief C. J., Ortega M., (2011), Satellite-based monitoring system for ionospheric disturbances, Retrieved from [https://www.researchgate.net/publication/265922592\\_Satellite-Based\\_Monitoring\\_System\\_for\\_Ionospheric\\_Disturbances](https://www.researchgate.net/publication/265922592_Satellite-Based_Monitoring_System_for_Ionospheric_Disturbances)
- 17) Indolia S., Goswami A. K., Mishra S.P., Asopa P., (2018), Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach, *Procedia Computer Science*, Volume 132, Pages 679-688, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2018.05.069>
- 18) Islam M., Chen G., Jin S., (2019), An overview of neural network, *American Journal of Neural Networks and Applications*, 5(1), 5, <https://doi.org/10.11648/j.ajnna.20190501.12>
- 19) Jähne B., (2005), Digitization, sampling, quantization in Digital image processing (pp. 111–142), Springer, Berlin, Heidelberg, [https://doi.org/10.1007/3-540-27563-0\\_9](https://doi.org/10.1007/3-540-27563-0_9)
- 20) Jocher G., (2024), YOLOv8 vs YOLOv5: A Detailed Comparison, Ultralytics YOLO Docs Retrieved from <https://docs.ultralytics.com/compare/yolov8-vs-yolov5/#other-yolo-models-to-consider>
- 21) Kirkby G., (2021), Lightning risks on mining operations, Lightningman Pty Ltd t/as Lightning and Surge Technologies, Specialist in lightning risk mitigation
- 22) Kithil R., (2014), Mining lightning safety in the mining industry, Technical Paper NLSI 06-14
- 23) KlearStack, (2023), AI image processing: Everything you need to know, Medium, <https://medium.com/@klear-stack/ai-image-processing-everything-you-need-to-know-4cd337e0149a>

- 24) Kumar T., Verma K, (2010), A Theory Based on Conversion of RGB image to Gray image, *International Journal of Computer Applications*, 7(2), 7-10
- 25) Lai J., Liu Y., Du J., Li Q., (2019), Lightning Detection Technology and Application, *International Conference on Meteorology Observations (ICMO)*, Chengdu, China, 2019, pp. 1-5, doi: 10.1109/ICMO49322.2019.9026107
- 26) Liu H., Sun F., Gu J., Deng L., (2022), SF-YOLOv5: A Lightweight Small Object Detection Algorithm Based on Improved Feature Fusion Mode, *Sensors*, 22(15), 5817, <https://doi.org/10.3390/s22155817>
- 27) Low R., Zhou J., (2022), Assessing the risk of operating mining equipment during lightning storms (Project No. D04489-001), *Resources Safety and Health Queensland*, Prepared for Mr. Neville Atkinson, Inspector of Mines - Electrical
- 28) Maggio C. R., Marshall T. C., Stolzenburg M., (2009), Estimations of charge transferred and energy released by lightning flashes in short bursts, *Journal of Geophysical Research: Atmospheres*, 114 (D14): D14203, doi:10.1029/2008JD011506
- 29) Martin A. U., (1984), *Lightning*, Dover Publications, Inc., New York, United States
- 30) Mata Viejo D., (2020), From 0 to Convolutional Neural Networks. Retrieved from <https://www.linkedin.com>
- 31) Meredith S. L., Earles S. K., (2010), Electromagnetic fields induced by lightning striking a perfect and imperfect conductive surface, *Proceedings of the 21st International Lightning Detection Conference*, Orlando, FL
- 32) NASA Science, *Science News*, (2001), Where Lightning Strikes, Retrieved from [science.nasa.gov/science-news/science-at-nasa/2001/ast05dec\\_1/](https://science.nasa.gov/science-news/science-at-nasa/2001/ast05dec_1/)
- 33) National Oceanic and Atmospheric Administration, (n.d.), How lightning is created, NOAA JetStream, Retrieved from <https://www.noaa.gov/jetstream/lightning/how-lightning-is-created>
- 34) Nguyen D. T., Deegan G., D'Alessandro F., (2001), Fractal nature of probabilistic model of lightning discharge, In *Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology (TENCON)* (Vol. 2), <https://doi.org/10.1109/TENCON.2001.949706>
- 35) Padmavathi K., Thangadurai K., (2016), Implementation of RGB and grayscale images in plant leaves disease detection—comparative study, *Indian Journal of Science and Technology*, 9(6), 1-6

- 36) Petrou M. M., Petrou C., (2010), Image processing: the fundamentals, John Wiley & Sons
- 37) Price C., (2009), Thunderstorms, Lightning and Climate Change, In: Betz, H.D., Schumann U., Laroche P. (eds) Lightning: Principles, Instruments and Applications. Springer, Dordrecht, [https://doi.org/10.1007/978-1-4020-9079-0\\_24](https://doi.org/10.1007/978-1-4020-9079-0_24)
- 38) Pritha A., Jeslet D. S., (2015), A study on sampling and quantization techniques of image processing, International Journal of Trend in Research and Development (IJTRD), Special Issue: National Conference on Prominent Challenges in Information Technology (PCIT), 8th Dec 2015, ISSN: 2394-9333
- 39) Raheem B., Ogbuju E., Oladipo, F, (2023), Development of a Lightning Prediction Model Using Machine Learning Algorithm: Survey, Journal of Applied Artificial Intelligence, 4(1), 45–56. <https://doi.org/10.48185/jaai.v4i1.727>
- 40) Rakov A., Vladimir A., Uman, Martin A., (2003), Lightning: Physics and effects, Cambridge, England, Cambridge University Press, ISBN 978-0521583275
- 41) Rakov V. A., Uman, M. A, (2003), Lightning: Physics and effects, Cambridge University Press, DOI: 10.1017/CBO9781107340886
- 42) Rakov V. A., (2016), Fundamentals of Lightning, Cambridge University Press
- 43) Raunek, (2021), How ships are protected from lightning – Ships earthing system, Marine Safety, Retrieved from <https://www.marineinsight.com/marine-safety/how-ships-are-protected-from-lightning-ships-earthing-system>
- 44) Rimboud A., Farges T., C-Labonnote L., Dubuisson P., Barthe C., Thieuleux F., (2024), 3D Monte-Carlo simulations of lightning optical waveforms and images observable by on-board operational instruments, Journal of Quantitative Spectroscopy and Radiative Transfer, ISSN 0022-4073, <https://doi.org/10.1016/j.jqsrt.2024.108950>
- 45) Ruman, (2023), YOLO Data augmentation explained, Medium, [YOLO Data Augmentation Explained](#)
- 46) Russell S. J., Norvig P., (2010), Artificial intelligence: A modern approach (3rd ed.), Upper Saddle River, NJ: Prentice Hall
- 47) Saha S., (2018), A comprehensive guide to convolutional neural networks — the ELI5 way, Towards Data Science, Retrieved from <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

- 48) Saravanan C., (2010), Color Image to Grayscale Image Conversion, Second International Conference on Computer Engineering and Applications, Bali, Indonesia, 2010, pp. 196-199, doi: 10.1109/ICCEA.2010.192
- 49) Sargoytchev S. S., (2019), Analysis of natural lightning phenomena and method for laboratory study, Retrieved from <https://vixra.org/abs/1911.0207>
- 50) Sarrel M., (2024), CPU vs. GPU: What's best for machine learning? Aerospike, Retrieved from <https://www.aerospike.com/blog/cpu-vs-gpu-whats-best-for-machine-learning>
- 51) Singhaniya R., (2021), Explain about image sampling and quantization process, Bench Partner, Retrieved from <https://benchpartner.com/q/explain-about-image-sampling-and-quantization-process>
- 52) Sithik M., (2023), My experiments with Yolov5: Almost everything you want to know about Yolov5-Series - Part 2, Medium, Retrieved from <https://medium.com>
- 53) Stracqualursi E., Datsios Z. G., Araneo R., Mikropoulos P. N., Tsovilis T. E., (2023), Finite Difference Time Domain Implementation of Leader Development Models for Lightning Studies, IEEE International Conference on Environment and Electrical Engineering and 2023 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), Madrid, Spain, pp. 1-6, doi: 10.1109/EEEIC/ICPSEurope57605.2023.10194799
- 54) Taylor A. R., (1971), Ecological aspects of lightning in forests, USDA Forest Service, Intermountain Forest and Range Experiment Station,. Ogden, UT. Stationed at Northern Forest Fire Laboratory, Missoula, MT
- 55) Thamali W., (2020), Sampling & quantization in digital image processing, Medium, Retrieved from <https://medium.com>
- 56) Thiemann E. M., Gasiewski A. J., (2014), Time-domain solution to Maxwell's equations for a lightning dart leader and subsequent return stroke, 2014 International Conference on Lightning Protection (ICLP), 6929566, <https://doi.org/10.1109/ICLP.2014.6973126>
- 57) Tishan Y., (2023), Understanding the difference between supervised and unsupervised learning techniques, <https://doi.org/10.13140/RG.2.2.36176.48641>
- 58) Timilsina A., (2024), YOLOv8 architecture explained!, Medium, Retrieved from <https://abintimilsina.medium.com/yolov8-architecture-explained-a5e90a560ce5>

- 59) Tsang S.-H., (2023), Brief review: YOLOv5 for object detection: Brief explanation of YOLOv5, Medium, <https://medium.com/@sik-ho-tsang/brief-review-yolov5-for-object-detection-it-outperforms-efficientdet-7f8b7fa8ffdf>
- 60) Uman M. A., (1986), All About Lightning, Dover Publications, Inc. pages 103–110, ISBN 978-0-486-25237-7
- 61) Uman M. A., (2001), The lightning discharge, Dover Publications
- 62) Whaley C., Etten-Bohm M., Schumacher C., et al, (2024), A new lightning scheme in the Canadian Atmospheric Model (CanAM5.1): implementation, evaluation, and projections of lightning and fire in future climates, Geoscientific Model Development, 17(18), pp.7141-7155. DOI: 10.5194/gmd-17-7141-2024
- 63) Xu P., Wang J., Jiang Y., Gong X., (2024), Applications of artificial intelligence and machine learning in image processing, *Frontiers in Materials*, <https://doi.org/10.3389/fmats.2024.1431179>
- 64) Xu R., Lin H., Lu K., Liu Y., (2021), A forest fire detection system based on ensemble learning, *Forests*, 12(2), 217, <https://doi.org/10.3390/f12020217>
- 65) Yang Z. R., (2006), A novel radial basis function neural network for discriminant analysis, *IEEE Transactions on Neural Networks*, 17(3), 604–612, doi:10.1109/tnn.2006.873282
- 66) Yanoviak S. P., Gora E., Gutierrez C., Bitzer P. M., (2024), The ecological effects of lightning in a tropical forest, In *The first 100 years of research on Barro Colorado: Plant and ecosystem science* (DOI: 10.5479/si.26880802), License: CC BY 4.0
- 67) Zhang X., Wang D., (2019), Application of artificial intelligence algorithms in image processing, *Journal of Visual Communication and Image Representation*, Volume 61, Pages 42-49, ISSN 1047-3203, <https://doi.org/10.1016/j.jvcir.2019.03.004>
- 68) Zhang Z., Tian Y., Wang G., Zheng C., Zhao F., (2024), A forest fire prediction method for lightning stroke based on remote sensing data, <https://doi.org/10.3390/f15040647>
- 69) Βλαχάβας Ι., Κεφαλάς Π., Βασιλειάδης Ν., Σακελλαρίου Ι., Κεφαλάς Π., (2002), Τεχνητή νοημοσύνη (1η έκδ.), Γαρταγάνης, ISBN 960-7013-28-X
- 70) Παλληκάρης Η., (2023), Νευρωνικά δίκτυα και εφαρμογές αυτών για την πρόβλεψη χρονοσειρών στο χρηματιστήριο, Master's thesis, Πανεπιστήμιο Πειραιώς
- 71) Περογιαννάκη Χ., ταξίαρχου Π.Α. τ. Γενικού Διευθυντού ΕΜΥ, Ναυτική Μετεωρολογία, Ίδρυμα Ευγενίδου 1974, Αθήνα

## ΠΑΡΑΡΤΗΜΑ

### Κώδικας 1: Μετατροπή εικόνων σε διαστάσεις 640x640

```
import cv2
import os

input_folder = r"C:\train_data\images\rr"
output_folder = r"C:\train_data\images\train_resized"
img_size = (640, 640)

os.makedirs(output_folder, exist_ok=True)

for img_name in os.listdir(input_folder):
    img_path = os.path.join(input_folder, img_name)

    if not img_name.lower().endswith((''.png', '.jpg', '.jpeg', '.bmp', '.tiff', '.webp', '.jfif')):
        print(f"Non-image file: {img_name}")
        continue

    img = cv2.imread(img_path)

    if img is None:
        print(f"OpenCV did not read: {img_path}")
        continue

    resized_img = cv2.resize(img, img_size)

    new_img_name = os.path.splitext(img_name)[0] + ".jpg"
    output_path = os.path.join(output_folder, new_img_name)

    cv2.imwrite(output_path, resized_img, [cv2.IMWRITE_JPEG_QUALITY, 95])

    print(f"Image resized and saved as JPEG: {output_path}")

print(" Images succesfully resized and saved as JPEG.")
```

### Κώδικας 2: Υπολογισμός τιμής hash



```

import os
import cv2
import hashlib

def image_hash(img_path):

    img = cv2.imread(img_path)
    if img is None:
        return None

    img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    img_resized = cv2.resize(img_gray, (8, 8))

    img_hash = hashlib.md5(img_resized.tobytes()).hexdigest()

    return img_hash

def find_duplicate_images(folder_path):

    image_hashes = { }
    duplicate_images = []

    for img_name in os.listdir(folder_path):
        img_path = os.path.join(folder_path, img_name)

        if not img_name.lower().endswith(('png', 'jpg', 'jpeg', 'bmp', 'tiff', 'jfif', 'webp')):
            continue

        img_hash = image_hash(img_path)

        if img_hash:

            if img_hash in image_hashes:

                duplicate_images.append((img_name, image_hashes[img_hash]))
            else:

                image_hashes[img_hash] = img_name

    return duplicate_images

folder_path = r"C:\train_data\images\val" #The folder path
duplicates = find_duplicate_images(folder_path)

if duplicates:
    print("Διπλότυπες εικόνες:")
    for img1, img2 in duplicates:
        print(f"Duplicate images: {img1} and {img2}")
else:
    print("Δεν εντοπίστηκαν διπλότυπες εικόνες.")

```

### Κώδικας 3: Υπολογισμός συνολικού όγκου δεδομένων

```
import os

folder = r"C:\train_data\train\images"
total_size = 0
file_count = 0

for file in os.listdir(folder):
    file_path = os.path.join(folder, file)

    if os.path.isfile(file_path):
        total_size += os.path.getsize(file_path)
        file_count += 1

avg_size = total_size / file_count if file_count > 0 else 0

print(f"Σύνολο εικόνων: {file_count}")
print(f"Συνολικό μέγεθος: {total_size / (1024 * 1024):.2f} MB")
print(f"Μέσο μέγεθος ανά εικόνα: {avg_size / 1024:.2f} KB")
```