

**ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ**  
**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ Η/Υ**  
Τομέας Ηλεκτρονικής & Τεχνολογίας Συστημάτων Πληροφορικής  
Εργαστήριο Αρχιτεκτονικής Υπολογιστών και Συστημάτων Υψηλών  
Επιδόσεων



**Συνελικτικά νευρωνικά δίκτυα για αυτόνομη οδήγηση**

Διπλωματική Εργασία

**Τερζόγλου Γεώργιος, ΑΜ 56713**

Επιβλέπων: ΕΔΙΠ Δρ. Ηρακλής Σπηλιώτης

Ξάνθη 2021



ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΡΑΚΗΣ

ΤΜΗΜΑ  
ΗΜ & ΜΥ



## ΠΕΡΙΛΗΨΗ

---

Η αυτόνομη οδήγηση αναπτύσσεται με ραγδαία ταχύτητα και αρχίζει να μπαίνει για τα καλά στη ζωή μας. Έχουν αρχίσει ήδη να είναι διαθέσιμα στην αγορά τα πρώτα αυτόνομα οχήματα. Επιστήμονες απ' όλο τον κόσμο ασχολούνται καθημερινά με την ανάπτυξη νέων και τη βελτίωση ήδη υπάρχοντων συστημάτων αυτόνομης οδήγησης. Πλέον, όλα τα οχήματα διαθέτουν συστήματα υποβοήθησης οδήγησης που κάνουν την οδήγηση πιο εύκολη και πιο ξεκούραστη για τον οδηγό.

Στην παρούσα διπλωματική εργασία, αφού γίνει μια εισαγωγή στο πεδίο της υπολογιστικής όρασης και στον τρόπο που τα νευρωνικά δίκτυα συμβάλλουν στην εξέλιξη αυτού του τομέα, θα αναλύσουμε μερικά βασικά προβλήματα που έχουν να αντιμετωπίσουν οι επιστήμονες που ασχολούνται με τη δημιουργία αυτόνομων οχημάτων και θα δώσουμε λύσεις έχοντας ως δεδομένα μόνο εικόνες που παίρνουμε κατά τη διάρκεια της οδήγησης. Τα προβλήματα που θα αναλυθούν είναι ο υπολογισμός των αποστάσεων των αντικειμένων και οχημάτων από το όχημά μας, η αναγνώριση της κυκλοφοριακής σήμανσης και των φωτεινών σηματοδοτών και η εύρεση της περιοχής που επιτρέπεται να οδηγήσει το όχημά μας για την παροχή συμβουλών πλοήγησης ή την αυτόνομη οδήγηση.

Για τον υπολογισμό των αποστάσεων θα πρέπει αρχικά να γίνει ο εντοπισμός των αντικειμένων και έπειτα να υπολογίσουμε την απόσταση τους από την κάμερα. Ο εντοπισμός των αντικειμένων έγινε με τη βοήθεια ενός προ-εκπαιδευμένου μοντέλου YOLO v5 της Ultralytics το οποίο μπορεί να εντοπίζει 80 διαφορετικά αντικείμενα. Ενώ για τον υπολογισμό της απόστασης χρησιμοποιήθηκε μια παραλλαγή του τύπου του εστιακού μήκους.

Η επίλυση του προβλήματος των πινακίδων χωρίζεται σε δύο μέρη, στον εντοπισμό τους ο οποίος έγινε με την εκπαίδευση ενός δικτύου YOLO v5 και την αναγνώρισή τους που επιτεύχθηκε με την ανάπτυξη ενός νευρωνικού δικτύου με συνελκτικά στρώματα. Για τους φωτεινούς σηματοδότες, έγινε χρήση της συνάρτησης HoughCircles της βιβλιοθήκης του OpenCV για τον εντοπισμό κύκλων σε εικόνες φωτεινών σηματοδοτών.

Τέλος, η εύρεση της περιοχής που επιτρέπεται να οδηγήσει το όχημά μας, έγινε με δύο τρόπους. Ο ένας είναι απλός και βασίζεται στην τεχνική εύρεσης γραμμών με τη χρήση της συνάρτησης HoughLinesP του OpenCV και ο άλλος στοχεύει στην επίτευξη τμηματικής κατηγοριοποίησης της εικόνας με τη χρήση βαθιάς μάθησης.

Η υλοποίηση έγινε με τη χρήση της γλώσσας προγραμματισμού Python στο περιβάλλον του Google Colab και των βιβλιοθηκών που χρησιμοποιούνται για νευρωνικά δίκτυα όπως η Keras, η TensorFlow η NumPy και η Pandas. Η επεξεργασία των εικόνων και η απεικόνιση των αποτελεσμάτων έγινε με τη χρήση των βιβλιοθηκών OpenCV και Matplotlib.

# ΠΕΡΙΕΧΟΜΕΝΑ

---

ΚΕΦΑΛΑΙΟ 1ο ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ .....	9
1.1 ΕΙΣΑΓΩΓΗ.....	9
1.2 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ.....	9
1.2.1 Επιβλεπόμενη Μάθηση (Supervised Learning) .....	10
1.2.2 Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning) .....	12
ΚΕΦΑΛΑΙΟ 2ο ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ .....	14
2.1 ΕΙΣΑΓΩΓΗ ΣΤΗ ΒΑΘΙΑ ΜΑΘΗΣΗ.....	14
2.2 ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΤΕΧΝΗΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ.....	15
2.2.1 Νευρωνικό Δίκτυο Εμπρόσθιας Τροφοδότησης Ενός Επιπέδου ( Single-Layer FNN) ..	15
2.2.2 Νευρωνικό Δίκτυο Εμπρόσθιας Τροφοδότησης Πολλών Επιπέδων (Multi-Layer FNN)	16
2.3 ΣΥΝΑΡΤΗΣΕΙΣ ΕΝΕΡΓΟΠΟΙΗΣΗΣ (ACTIVATION FUNCTIONS) .....	17
2.3.1 Γραμμική Συνάρτηση ενεργοποίησης (Linear Function) .....	18
2.3.2 Βηματική Συνάρτηση Ενεργοποίησης (Binary Step) .....	19
2.3.3 Σιγμοειδής Συνάρτηση Ενεργοποίησης (Sigmoid Function) .....	19
2.3.4 Συνάρτηση Υπερβολικής Εφαπτομένης (tanh Function) .....	19
2.3.5 Ανορθωμένη Γραμμική Συνάρτηση Ράμπας (Rectified Linear Unit Function) .....	20
2.3.6 Διακεκομμένη Ανορθωμένη Γραμμική Συνάρτηση Ράμπας (leaky-ReLU).....	21
2.3.7 Συνάρτηση Ενεργοποίησης Softmax .....	22
2.4 ΣΥΝΑΡΤΗΣΕΙΣ ΣΦΑΛΜΑΤΟΣ (LOSS FUNCTIONS) .....	22
2.4.1 Μέσο Τετραγωνικό Σφάλμα (Mean Squared Error) .....	22
2.4.2 Συνάρτηση Σφάλματος Binary Cross-Entropy .....	23
2.4.3 Συνάρτηση Σφάλματος Categorical Cross-entropy .....	23
2.4.4 Συνάρτηση Σφάλματος Poisson.....	23
2.5 ΑΛΓΟΡΙΘΜΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ (OPTIMIZATION ALGORITHMS) .....	24
2.5.1 Αλγόριθμος Ομαδικής Κατάβασης Κλίσης (Batch Gradient Descent) .....	24
2.5.2 Στοχαστικός Αλγόριθμος Κατάβασης Κλίσης (Stochastic Gradient Descent) .....	26
2.5.3 Αλγόριθμος Κατάβασης Κλίσης Μικρής Παρτίδας (Mini-Batch Gradient Descent).....	26
2.5.4 Αλγόριθμος Βελτιστοποίησης Adam .....	26
2.6 ΑΛΓΟΡΙΘΜΟΣ ΟΠΙΣΘΟΔΙΑΔΟΣΗΣ (BACKPROPAGATION ALGORITHM) .....	27

ΚΕΦΑΛΑΙΟ 3ο ΣΥΝΕΛΙΚΤΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ (CNN).....	29
3.1 ΕΙΣΑΓΩΓΗ.....	29
3.2 ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΤΩΝ ΣΥΝΕΛΙΚΤΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ .....	30
3.2.1 Συνελικτικά Επίπεδα Νευρώνων (Convolutional Layers).....	30
3.2.2 Δειγματοληπτικά Επίπεδα ή Επίπεδα Ομαδοποίησης (Pooling Layers Or Sampling) .	34
3.2.3 Πλήρως Συνδεδεμένα Επίπεδα (Fully Connected Layers) .....	35
3.3 ΕΠΙΠΕΔΑ ΕΓΚΑΤΑΛΕΙΨΗΣ (DROPOUT LAYERS) .....	36
3.4 ΒΙΒΛΙΟΘΗΚΕΣ ΚΑΙ ΠΛΑΤΦΟΡΜΕΣ ΛΟΓΙΣΜΙΚΟΥ .....	38
3.4.1 Tensorflow.....	38
3.4.2 Keras .....	38
3.4.3 NumPy.....	39
3.4.4 OpenCV .....	39
3.5 ΥΛΟΠΟΙΗΣΗ ΣΥΝΕΛΙΚΤΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΜΕ GPU .....	39
ΚΕΦΑΛΑΙΟ 4ο ΕΝΤΟΠΙΣΜΟΣ ΑΝΤΙΚΕΙΜΕΝΩΝ ΚΑΙ ΥΠΟΛΟΓΙΣΜΟΣ ΑΠΟΣΤΑΣΕΩΝ .....	41
4.1 ΕΙΣΑΓΩΓΗ .....	41
4.2 ΑΛΓΟΡΙΘΜΟΣ YOLO (YOU ONLY LOOK ONCE) .....	42
4.3 ΕΝΤΟΠΙΣΜΟΣ ΑΝΤΙΚΕΙΜΕΝΩΝ.....	44
4.4 ΥΠΟΛΟΓΙΣΜΟΣ ΑΠΟΣΤΑΣΕΩΝ .....	46
ΚΕΦΑΛΑΙΟ 5ο ΑΝΑΓΝΩΡΙΣΗ ΚΥΚΛΟΦΟΡΙΑΚΗΣ ΣΗΜΑΝΣΗΣ ΚΑΙ ΦΩΤΕΙΝΩΝ ΣΗΜΑΤΟΔΟΤΩΝ .....	51
5.1 ΕΙΣΑΓΩΓΗ.....	51
5.2 ΤΑΞΙΝΟΜΗΣΗ ΚΥΚΛΟΦΟΡΙΑΚΗΣ ΣΗΜΑΝΣΗΣ .....	51
5.2.1 Σύνολο Δεδομένων (Dataset).....	52
5.2.2 Προ-Επεξεργασία Δεδομένων .....	53
5.2.3 Εκπαίδευση Δικτύου .....	54
5.2.4 Αποτελέσματα ελέγχου .....	57
5.3 ΕΝΤΟΠΙΣΜΟΣ ΚΥΚΛΟΦΟΡΙΑΚΗΣ ΣΗΜΑΝΣΗΣ .....	60
5.3.1 Σύνολο Δεδομένων (Dataset).....	61
5.3.2 Προ-Επεξεργασία Δεδομένων .....	63
5.3.3 Εκπαίδευση Δικτύου .....	64
5.4 ΣΥΝΔΥΑΣΜΟΣ ΤΩΝ ΔΙΚΤΥΩΝ.....	65
5.5 ΑΝΑΓΝΩΡΙΣΗ ΦΩΤΕΙΝΟΥ ΣΗΜΑΤΟΔΟΤΗ.....	67
ΚΕΦΑΛΑΙΟ 6ο ΕΥΡΕΣΗ ΕΠΙΤΡΕΠΤΗΣ ΠΕΡΙΟΧΗΣ ΟΔΗΓΗΣΗΣ.....	72
6.1 ΕΙΣΑΓΩΓΗ.....	72

6.2 ΔΙΑΤΗΡΗΣΗ ΤΟΥ ΟΧΗΜΑΤΟΣ ΣΤΟ ΜΕΣΟ ΤΗΣ ΛΩΡΙΔΑΣ ΟΔΗΓΗΣΗΣ.....	72
6.2.1 Ανίχνευση Των Γραμμών Της Λωρίδας Οδήγησης .....	72
6.2.2 Συμβουλές Πλοήγησης .....	75
6.3 ΤΜΗΜΑΤΙΚΗ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ (SEMANTIC SEGMENTATION) .....	77
6.4 ΕΥΡΕΣΗ ΕΠΙΤΡΕΠΤΗΣ ΠΕΡΙΟΧΗΣ ΟΔΗΓΗΣΗΣ .....	79
6.4.1 Σύνολο Δεδομένων (Dataset) .....	79
6.4.2 Προ-Επεξεργασία Δεδομένων .....	80
6.4.3 Εκπαίδευση Δικτύου .....	81
6.4.4 Αποτελέσματα Ελέγχου .....	82
ΚΕΦΑΛΑΙΟ 7 <sup>ο</sup> ΣΥΝΟΨΗ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΚΑΤΕΥΘΥΝΣΕΙΣ .....	86
7.1 ΣΥΝΟΨΗ .....	86
7.2 ΜΕΛΛΟΝΤΙΚΕΣ ΚΑΤΕΥΘΥΝΣΕΙΣ .....	87
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	89

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

---

Σχήμα 1.1 Βασική Δομή Διαδικασίας Επιβλεπόμενης Μάθησης .....	10
Σχήμα 1.2 Παράδειγμα Προβλήματος Ταξινόμησης .....	11
Σχήμα 1.3 Παράδειγμα Προβλήματος Παλινδρόμησης.....	11
Σχήμα 1.4 Έννοια Ομαδοποίησης .....	12
Σχήμα 1.5 Βασική Δομή Διαδικασίας Ενισχυτικής Μάθησης.....	13
Σχήμα 2.1 AI, ML και Deep Learning .....	14
Σχήμα 2.2 Νευρωνικό Δίκτυο Εμπρόσθιας Τροφοδότησης .....	16
Σχήμα 2.3 Αρχιτεκτονική Αναδρομικού Νευρωνικού Δικτύου .....	17
Σχήμα 2.4 Γραμμική Συνάρτηση Ενεργοποίησης .....	18
Σχήμα 2.5 2.3.3 Σιγμοειδής Συνάρτηση Ενεργοποίησης.....	19
Σχήμα 2.6 Συνάρτηση Υπερβολικής Εφαπτομένης .....	20
Σχήμα 2.7 Ανορθωμένη Γραμμική Συνάρτηση Ράμπας .....	21
Σχήμα 2.8 Διακεκομμένη Ανορθωμένη Γραμμική Συνάρτηση Ράμπας .....	21
Σχήμα 2.9 Αλγόριθμος Ομαδικής Κατάβασης Κλίσης.....	25
Σχήμα 3.1 Δομή Συνελικτικών νευρωνικών δικτύων .....	30
Σχήμα 3.2 Εφαρμογή Συνελικτικού Φίλτρου Σε Ασπρόμαυρη Εικόνα .....	31
Σχήμα 3.3 Εφαρμογή Συνελικτικού Φίλτρου Σε Έγχρωμη Εικόνα .....	32
Σχήμα 3.4 Μηδενική επένδυση (Padding) .....	33
Σχήμα 3.5 Φίλτρο Μέγιστης Ομαδοποίησης.....	35
Σχήμα 3.6 Φίλτρο Μέσης Ομαδοποίησης.....	35
Σχήμα 3.7 Flattening .....	36
Σχήμα 3.8 Υπερεξειδίκευση Και Ανεπαρκείς Εξειδίκευση (Overfitting And Underfitting) .....	37
Σχήμα 4.1 Διαφορά Μεταξύ Ταξινόμησης Εικόνων Και Εύρεσης Αντικειμένων.....	41
Σχήμα 4.2 Αξιολόγηση Πρόβλεψης Με Βαθμολογία IOU .....	43
Σχήμα 4.3 Εξαγωγή Προβλέψεων Με Χρήση Του Αλγορίθμου YOLO .....	43
Σχήμα 4.4 Δομή Του Δικτύου YOLO .....	44
Σχήμα 4.5 Οι 80 Κλάσεις Του Σετ Δεδομένων COCO .....	45
Σχήμα 4.6 Εφαρμογή Προεκπαιδευμένου Μοντέλου YOLO .....	45
Σχήμα 4.7 Αρχείο .txt Εξόδου Του YOLO .....	46
Σχήμα 4.8 Εικόνα Απεικόνισης Αντικειμένων Με Γνωστή Απόσταση Από Τον Φακό .....	47



Σχήμα 4.9 Εντοπισμός Αντικειμένων Στο Σχήμα 4.8 .....	47
Σχήμα 4.10 Παράδειγμα Εικόνας Εισόδου Υπολογισμού Αποστάσεων .....	48
Σχήμα 4.11 Παράδειγμα Εικόνας Εξόδου Υπολογισμού Αποστάσεων .....	49
Σχήμα 4.12 Λόγος Εσφαλμένης Εκτίμησης της Απόστασης .....	50
Σχήμα 5.1 Οι 43 Κλάσεις του GTSRB .....	52
Σχήμα 5.2 Δεδομένα Εκπαίδευσης Ανά κλάση.....	53
Σχήμα 5.3 Διαστάσεις Εικόνων .....	53
Σχήμα 5.4 Αποτελέσματα Εφαρμογής Data Augmentation .....	54
Σχήμα 5.5 Το Summary Του Μοντέλου Αναγνώρισης Των Πινακίδων .....	55
Σχήμα 5.6 Αποτελέσματα Εκπαίδευσης.....	56
Σχήμα 5.7 Επιτυχημένη Πρόβλεψη Κλάσης Πινακίδας.....	56
Σχήμα 5.8 Ο Confusion Matrix Του Δικτύου .....	58
Σχήμα 5.9 Το Classification_Report Του Δικτύου .....	59
Σχήμα 5.10 Παραδείγματα Εικόνων Του Συνόλου Δεδομένων Εντοπισμού Κυκλοφοριακής Σήμανσης.....	61
Σχήμα 5.11 Παράδειγμα Αρχείου .txt Του Συνόλου Δεδομένων Εντοπισμού Κυκλοφοριακής Σήμανσης.....	61
Σχήμα 5.12 Οι 43 Κλάσεις του Συνόλου Δεδομένων Εντοπισμού Κυκλοφοριακής Σήμανσης .....	62
Σχήμα 5.13 Διαχωρισμός Των Κλάσεων του Συνόλου Δεδομένων Εντοπισμού Κυκλοφοριακής Σήμανσης Σε Υπό-κλάσεις .....	63
Σχήμα 5.15 Εντοπισμός Πινακίδων Σε Εικόνα .....	66
Σχήμα 5.14 Αποτελέσματα εκπαίδευσης δικτύου .....	65
Σχήμα 5.16 Παραγόμενο Αρχείο .txt Από Εντοπισμό .....	66
Σχήμα 5.17 Αποθήκευση Εικόνας Πινακίδας .....	67
Σχήμα 5.18 Αναγνώριση Πινακίδας .....	67
Σχήμα 5.19 Παραδείγματα Εφαρμογής Χρωματικών Φίλτρων Σε Εικόνες Φωτεινών Σηματοδοτών .....	69
Σχήμα 5.20 Πρόβλεψη Χρώματος Φωτεινών Σηματοδοτών.....	70
Σχήμα 5.21 Πρόβλεψη Διαφόρων Σχημάτων Φωτεινών Σηματοδοτών .....	71
Σχήμα 6.1 Εφαρμογή Της Συνάρτησης Canny Στην Εικόνα Εισόδου .....	73
Σχήμα 6.2 Μάσκα Περιοχής Ενδιαφέροντος.....	73
Σχήμα 6.3 Εικόνα Ακμών Της Περιοχής Ενδιαφέροντος .....	74

Σχήμα 6.4 Εύρεση Γραμμών .....	74
Σχήμα 6.5 Επέκταση Γραμμών .....	75
Σχήμα 6.6 Παραδείγματα Συμβουλών πλοήγησης Κατά Την Εύρεση Δύο Γραμμών .....	76
Σχήμα 6.7 Παραδείγματα Συμβουλών Πλοήγησης Κατά Την Εύρεση Μίας Γραμμής .....	77
Σχήμα 6.9 Σχηματική Απεικόνιση Του Δικτύου UNET.....	79
Σχήμα 6.8 Τμηματική Κατηγοριοποίηση Σε Εικόνες Δρόμου .....	78
Σχήμα 6.10 Παραδείγματα Εικόνων Επιτρεπτής Περιοχής Οδήγησης Του BDD100K.....	80
Σχήμα 6.11 Επεξεργασία Των Μασκών Του BDD100K .....	81
Σχήμα 6.12 Σχηματική Απεικόνιση Της Πορείας Της Εκπαίδευσης Του Δικτύου .....	82
Σχήμα 6.13 Πρόβλεψη Επιτρεπτής Περιοχής Οδήγησης .....	82
Σχήμα 6.14 Σύγκριση Εικόνας Πρόβλεψης Με Την Αναμενόμενη Εικόνα Εξόδου .....	83
Σχήμα 6.15 Ακρίβεια Προβλέψεων.....	83
Σχήμα 6.16 Προβλέψεις Δικτύου Σε Δύσκολες Εικόνες Δρόμου .....	85



# ΚΕΦΑΛΑΙΟ 1ο

---

## Τεχνητή νοημοσύνη

### 1.1 ΕΙΣΑΓΩΓΗ

Η τεχνητή νοημοσύνη είναι ένας ευρύς και πολύ διαδεδομένος στις μέρες μας κλάδος της μηχανικής και επιστήμης των υπολογιστών που ασχολείται με τη μελέτη και το σχεδιασμό ευφυών μηχανών, ικανών να εκτελούν εργασίες οι οποίες απαιτούν ανθρώπινη νοημοσύνη. Η τεχνητή νοημοσύνη συστηματοποιεί και αυτοματοποιεί τις διανοητικές εργασίες ενός ανθρώπινου εγκεφάλου. Για αυτόν το λόγο, μπορεί να εφαρμοστεί σε οποιαδήποτε πρόβλημα. Με αυτή την έννοια, είναι πράγματι ένα πολύ μεγάλο πεδίο.

Η τεχνητή νοημοσύνη χωρίζεται γενικά σε δύο μεγάλες κατηγορίες, τη συμβολική και την υπό-συμβολική νοημοσύνη [1]. Η συμβολική νοημοσύνη είναι η επιστήμη που μελετά την ανθρώπινη νοημοσύνη όσο και τον τρόπο αναπαραγωγής της σε υπολογιστές με τη χρήση συμβόλων για την εκτέλεση συγκεκριμένων εργασιών. Μερικά παραδείγματα αυτής είναι τα λογισμικά αναγνώρισης εικόνων, η έξυπνη αναζήτηση και η αυτόνομη οδήγηση. Αντιθέτως, η υπό-συμβολική νοημοσύνη μελετά την δημιουργία μιας μηχανής με ευφυΐα ανθρώπινου επιπέδου. Η ανάπτυξη αυτής δεν έχει προχωρήσει μέχρι σήμερα στον ίδιο βαθμό όσο η τεχνητή νοημοσύνη με τη χρήση συμβόλων.

### 1.2 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

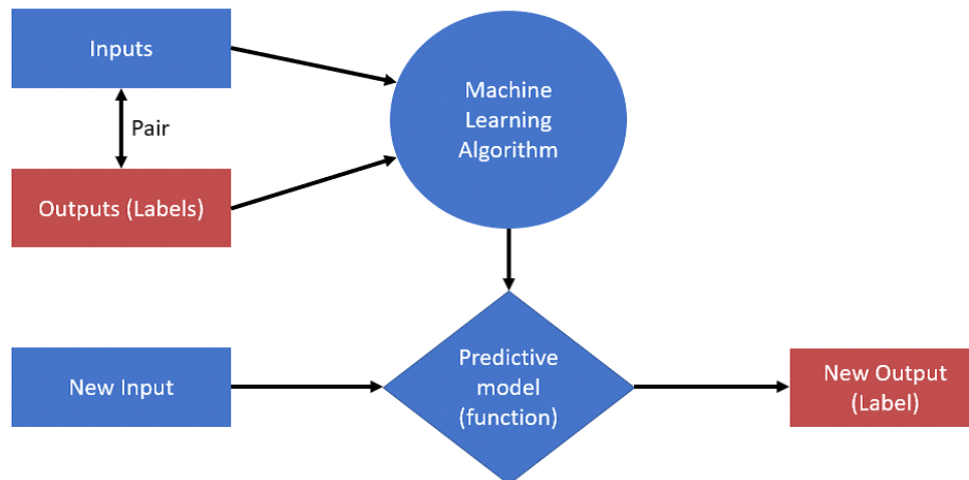
Η μηχανική μάθηση είναι η ανάπτυξη αλγορίθμων μάθησης για την απόκτηση γνώσεων από ένα σύνολο δεδομένων με σκοπό την δημιουργία μοντέλων που πραγματοποιούν προβλέψεις. Με τη χρήση αλγορίθμων μηχανικής μάθησης, αντί να απαιτείται από τους ανθρώπους να παράγουν χειροκίνητα τους κανόνες και να δημιουργούν δικά τους μοντέλα μετά από ανάλυση μεγάλου αριθμού δεδομένων, προσφέρεται μια πολύ πιο αποτελεσματική λύση για τη σταδιακή βελτίωση της απόδοσης των προγνωστικών μοντέλων σύμφωνα με τα δεδομένα.

Ζούμε σε μια εποχή όπου μεγάλες ποσότητες δεδομένων πάσης φύσεως είναι διαθέσιμες σε όλους. Χρησιμοποιώντας τους αλγόριθμους μηχανικής μάθησης, μπορούμε να μετατρέψουμε αυτά τα δεδομένα σε γνώση. Γι' αυτό το λόγο, η μηχανική μάθηση καθίσταται όλο και πιο σημαντική στην έρευνα της επιστήμης του μηχανικού υπολογιστών και συνδυάζεται όλο και πιο

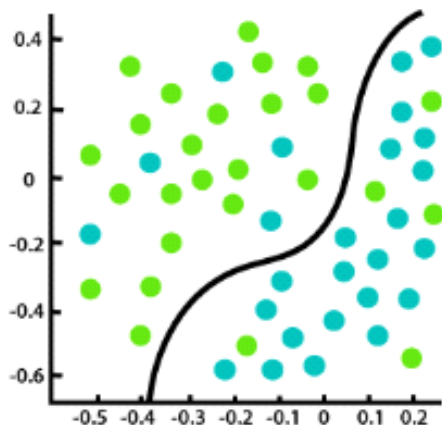
πολύ με άλλες επιστήμες, όπως για παράδειγμα της οικονομίας για τη δημιουργία μοντέλων πρόβλεψης της πορείας μιας μετοχής.

### 1.2.1 Επιβλεπόμενη Μάθηση (Supervised Learning)

Η επιβλεπόμενη μάθηση ή αλλιώς μάθηση υπό επίβλεψη είναι μία κατηγορία μηχανικής μάθησης με κύριο στόχο την εκπαίδευση ενός μοντέλου από ένα σύνολο δεδομένων με ετικέτα που μας επιτρέπει να κάνουμε προβλέψεις για μελλοντικά δεδομένα. Κάθε στοιχείο των δεδομένων δηλαδή είναι ένα ζεύγος που αποτελείται από την είσοδο είτε αυτή είναι μια απλή τιμή είτε μια εικόνα και την επιθυμητή τιμή εξόδου. Το μοντέλο «εκπαιδεύεται» σε αυτά τα δεδομένα και έτσι σε νέες εισόδους χωρίς έξοδο μπορεί να προβλέψει ποια είναι η αντίστοιχη έξοδος. Αυτή η δομή απεικονίζεται στο Σχήμα 1.1.



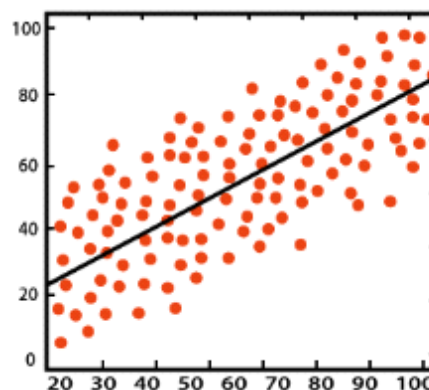
Σχήμα 1.1 Βασική Δομή Διαδικασίας Επιβλεπόμενης Μάθησης [9]



Σχήμα 1.2 Παράδειγμα Προβλήματος Ταξινόμησης [10]

Τα προβλήματα της επιβλεπόμενης μάθησης χωρίζονται σε δύο κύριες κατηγορίες, στα προβλήματα ταξινόμησης και τα προβλήματα παλινδρόμησης [2]. Σε περίπτωση προβλήματος ταξινόμησης, υπάρχει μια ξεχωριστή ομάδα εξόδων στις οποίες αντιστοιχεί ένα πρότυπο το οποίο μπορεί να εμφανισθεί στην είσοδο. Υπ' αυτή την έννοια, το πρότυπο στην είσοδο «ταξινομείται» όταν εισάγεται στο μοντέλο πρόβλεψης. Ένα παράδειγμα προβλήματος ταξινόμησης είναι η ανίχνευση καρδιακών παθήσεων. Θα μπορούσε μάλιστα να είναι πρόβλημα δυαδικής ταξινόμησης καθώς μπορεί να υπάρχουν μόνο δύο κατηγορίες, να έχουν καρδιακές παθήσεις ή όχι. Το μοντέλο ταξινόμησης, σε αυτήν την περίπτωση, χρειάζεται πολλά δεδομένα για να κατανοήσει πώς οι δεδομένες τιμές εισόδου σχετίζονται με την κάθε κλάση. Όταν εκπαιδευτεί με ακρίβεια, θα μπορεί να χρησιμοποιηθεί για να ανιχνεύσει εάν υπάρχει καρδιακή νόσος ή όχι για τον καθένα.

Στην περίπτωση προβλήματος παλινδρόμησης, μας δίνεται ένας αριθμός δεδομένων για είσοδο και μια μεταβλητή συνεχούς απόκρισης ως έξοδος και προσπαθούμε να βρούμε μια σχέση μεταξύ αυτών των μεταβλητών που μας επιτρέπουν να προβλέψουμε μια έξοδο. Ένα παράδειγμα προβλήματος παλινδρόμησης είναι η πρόβλεψη της βαθμολογίας των μαθητών για μια εξέταση. Ως δεδομένα εισόδου θα μπορούσε να έχει τον χρόνο μελέτης, το φύλο, την ηλικία, τον δείκτη νοημοσύνης και ως έξοδο την βαθμολογία του κάθε μαθητή. Το μοντέλο παλινδρόμησης μόλις ολοκληρώσει την εκπαίδευση θα είναι σε θέση να προβλέπει για τον κάθε μαθητή την βαθμολογία του από μηδέν ως εκατό ή ως δέκα (ανάλογα με την κλίμακα που θα βρίσκονται τα δεδομένα μας). Το ίδιο παράδειγμα, αν η έξοδος αντί για βαθμολογία ήταν ο προβιβασμός του μαθητή ή όχι, μετατρέπεται σε πρόβλημα ταξινόμησης.



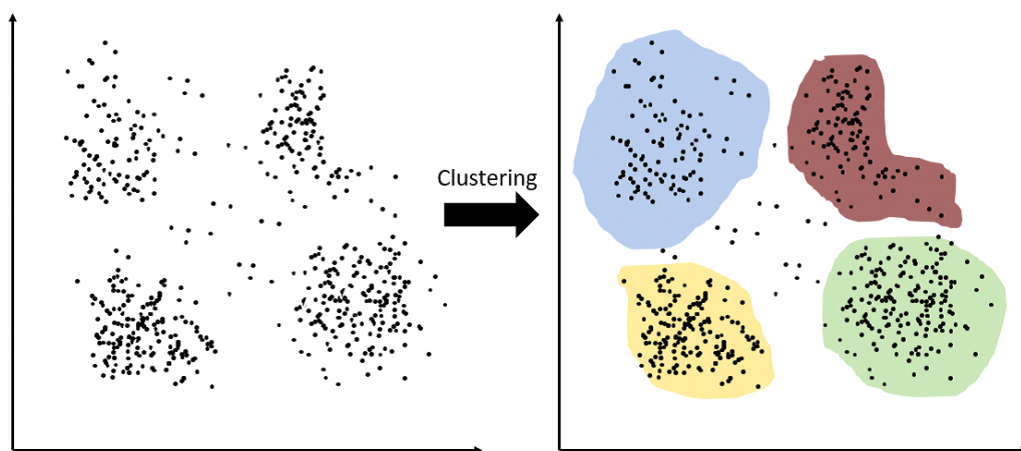
Σχήμα 1.3 Παράδειγμα Προβλήματος Παλινδρόμησης [10]

### 1.2.2 Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning)

Στη μη επιβλεπόμενη μάθηση έχουμε να κάνουμε με δεδομένα χωρίς ετικέτα ή δεδομένα άγνωστης δομής. Με τη χρησιμοποίηση τεχνικών μη επιβλεπόμενης μάθησης, είμαστε σε θέση να διερευνήσουμε τη δομή των δεδομένων μας και να εξάγουμε ουσιαστικές πληροφορίες δίχως την καθοδήγηση μιας γνωστής μεταβλητής εξόδου. Οι αλγόριθμοι σ' αυτή την περίπτωση δεν προσπαθούν να βρουν μια συγκεκριμένη απάντηση σε ένα πρόβλημα. Αντιθέτως, ο στόχος τους είναι να εντοπίσουν ποιοτικά πρότυπα και ομαδοποιήσεις μεταξύ των δεδομένων εισόδου που δεν μπορούν να προσδιοριστούν με απλή ανθρώπινη λογική σκέψη.

Η ομαδοποίηση, η πιο βασική λειτουργία της μη εποπτευόμενης μάθησης, είναι μια διερευνητική τεχνική ανάλυσης δεδομένων που μας δίνει τη δυνατότητα να οργανώσουμε πολλές πληροφορίες, δηλαδή δεδομένα εισόδου, σε σημαντικές υποομάδες χωρίς προηγούμενη γνώση παλαιότερων μελών της κάθε ομάδας [3]. Κάθε σύμπλεγμα που μπορεί να προκύψει κατά την διάρκεια της ανάλυσης ορίζει μια ομάδα αντικειμένων που μοιράζονται έναν συγκεκριμένο βαθμό ομοιότητας αλλά είναι πιο ανόμοια με αντικείμενα άλλων συμπλεγμάτων. Γι' αυτό, η ομαδοποίηση ονομάζεται και μη εποπτευόμενη ταξινόμηση. Η ομαδοποίηση είναι μια εξαιρετική τεχνική για τη δόμηση της διαθέσιμης πληροφορίας αντλώντας σχέσεις μεταξύ των δεδομένων. Για παράδειγμα, προσφέρει τη δυνατότητα στους εμπόρους να ανακαλύψουν ομάδες πελατών σύμφωνα με τα ενδιαφέροντά τους προκειμένου να αναπτύξουν διαφορετικά προγράμματα μάρκετινγκ για την κάθε ομάδα.

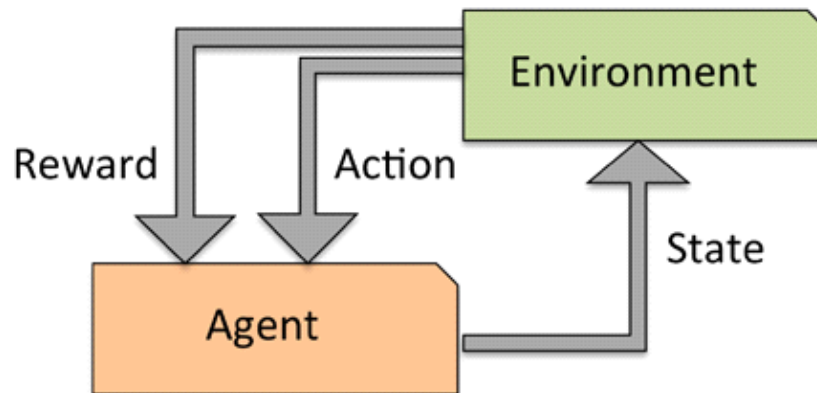
Το παρακάτω σχήμα μας δείχνει πώς η ομαδοποίηση μπορεί να εφαρμοστεί στην οργάνωση δεδομένων χωρίς ετικέτα, κάποια αντίστοιχη έξοδο δηλαδή για κάθε τιμή εισόδου, σε τέσσερις διαφορετικές ομάδες σύμφωνα με την ομοιότητα των χαρακτηριστικών τους.



Σχήμα 1.4 Έννοια Ομαδοποίησης [11]

### 1.2.3 Ενισχυτική Μάθηση (Reinforcement Learning)

Ένας άλλος τύπος μηχανικής μάθησης είναι η ενισχυτική μάθηση [4]. Η ενισχυτική μάθηση αποσκοπεί στη δημιουργία ενός συστήματος που βελτιώνει την απόδοσή του βασισμένο σε αλληλεπιδράσεις με το περιβάλλον. Η ενισχυτική μάθηση διαφέρει από άλλους τύπους μάθησης στο ότι το σύστημα δεν εκπαιδεύεται σε ένα σύνολο δεδομένων. Οι πληροφορίες για την τρέχουσα κατάσταση του περιβάλλοντος περιλαμβάνουν τη δράση και ένα σήμα ανταμοιβής που τροφοδοτούνται πίσω στο σύστημα. Το σύστημα μαθαίνει μέσα από την δοκιμή και το σφάλμα. Για κάθε δράση δηλαδή, όσο πιο θετικό είναι το σήμα ανταμοιβής τόσο πιο σωστή την θεωρεί για την συγκεκριμένη κατάσταση του περιβάλλοντος.



Σχήμα 1.5 Βασική Δομή Διαδικασίας Ενισχυτικής Μάθησης

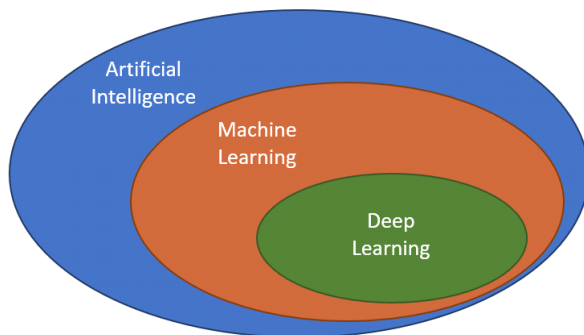
Ένα χαρακτηριστικό και πολύ δημοφιλές παράδειγμα ενισχυτικής μάθησης είναι μια μηχανή που παίζει σκάκι. Εδώ, η κατάσταση του περιβάλλοντος είναι ο τρόπος που είναι τοποθετημένα τα πιόνια στην σκακιέρα σε κάθε κίνηση, η ανταμοιβή της κάθε πιθανής σειράς κινήσεων μπορεί να οριστεί ως νίκη ή ήττα στο τέλος του παιχνιδιού. Με αυτά τα δεδομένα, το σύστημα σε κάθε κίνηση αποφασίζει ποια είναι η καλύτερη σειρά κινήσεων που πρέπει να ακολουθήσει ώστε να κερδίσει την παρτίδα.



## ΚΕΦΑΛΑΙΟ 2ο

### Τεχνητά νευρωνικά δίκτυα

#### 2.1 ΕΙΣΑΓΩΓΗ ΣΤΗ ΒΑΘΙΑ ΜΑΘΗΣΗ



Σχήμα 2.1 AI, ML και Deep Learning [12]

Ενώ η τεχνητή νοημοσύνη (Artificial Intelligence) είναι ένας πολύ ευρύς επιστημονικός τομέας που σχετίζεται με την αυτοματοποίηση των πνευματικών εργασιών που συνήθως εκτελούνται από ανθρώπους, η μηχανική μάθηση (Machine Learning) ασχολείται με την εύρεση χρήσιμων αναπαραστάσεων από ορισμένα δεδομένα εισόδου, με την χρήση ενός

βρόγχου ανατροφοδότησης και συνήθως υπάρχει ένα σύνολο κανόνων που πρέπει να τηρούν αυτές οι αναπαραστάσεις [5]. Όπως υποδηλώνεται στο Σχήμα 2.1 η βαθιά μάθηση (Deep Learning) είναι μια υποκατηγορία της μηχανικής μάθησης. Η βαθιά μάθηση μπορεί να γίνει κατανοητή ως ένα σύνολο αλγορίθμων που αναπτύχθηκαν για να εκπαιδεύσουν πολλά επίπεδα με πιο αποτελεσματικό τρόπο και μπορεί να είναι είτε με επίβλεψη είτε χωρίς. Η λέξη βαθιά υποδηλώνει ότι αντί για ένα ή δύο επίπεδα, όπως σε άλλες προσεγγίσεις της μηχανικής μάθησης, η οποία συχνά αναφέρεται ως ρηχή μάθηση, υπάρχουν πολλά διαδοχικά στρώματα περίπλοκων αναπαραστάσεων. Μια κατηγορία μοντέλων που διευκολύνουν τη μαθησιακή διαδικασία για τις αναπαραστάσεις αυτές είναι τα νευρωνικά δίκτυα.

Τα τεχνητά νευρωνικά δίκτυα εμπνεύστηκαν από την επεξεργασία πληροφοριών και τους κατακευκτισμένους κόμβους επικοινωνίας των βιολογικών συστημάτων. Αποτελούνται από διασυνδεδεμένα υπολογιστικά στοιχεία που διαθέτουν την ικανότητα να ανταποκρίνονται σε ερεθίσματα που δέχονται στην είσοδό τους και να μαθαίνουν να προσαρμόζονται στο περιβάλλον τους με στόχο την υπολογιστική προσομοίωση της λειτουργίας των βιολογικών νευρωνικών δικτύων σύμφωνα με κάποιο μαθηματικό μοντέλο τους.

## 2.2 ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΤΕΧΝΗΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ

Η βασική δομή των τεχνητών νευρωνικών δικτύων είναι ο νευρώνας [6]. Ο νευρώνας μπορεί να θεωρηθεί ως στοιχείο επεξεργασίας. Στα τεχνητά νευρωνικά δίκτυα οι νευρώνες συνδέονται μεταξύ τους μέσω των βαρών. Τα βάρη αντιπροσωπεύουν τη σημαντικότητα των συνδέσεων μεταξύ των νευρώνων. Εάν το βάρος από έναν νευρώνα Α προς έναν νευρώνα Β έχει μεγαλύτερο μέγεθος από τα βάρη των υπολοίπων, αυτό σημαίνει πως ο νευρώνας Α έχει μεγαλύτερη επιρροή στο Β από τους άλλους. Εάν ένας νευρώνας θα πρέπει να ενεργοποιηθεί ή όχι καθορίζεται από μια συνάρτηση ενεργοποίησης που διαθέτει ο καθένας. Ο σκοπός της λειτουργίας ενεργοποίησης είναι να εισάγει τη μη γραμμικότητα στην έξοδο ενός νευρώνα.

Με την πάροδο του χρόνου και την ραγδαία εξέλιξη της επιστήμης των τεχνητών νευρωνικών δικτύων, αυξήθηκαν σε μεγάλο βαθμό οι αρχιτεκτονικές νευρωνικών δικτύων που χρησιμοποιούνται. Η διαφορά τους βρίσκεται κυρίως στον τύπο σύνδεσης μεταξύ των νευρώνων. Οι διαφορετικές αρχιτεκτονικές χρησιμοποιούνται για την καλύτερη προσαρμογή των νευρωνικών δικτύων στο εκάστοτε πρόβλημα. Τις διαφορές και τη δομή τους θα παραθέσουμε στη συνέχεια.

### 2.2.1 Νευρωνικό Δίκτυο Εμπρόσθιας Τροφοδότησης Ενός Επιπέδου ( Single-Layer FNN)

Σε ένα νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης (FeedForward Neural Network), οι πληροφορίες κινούνται προς μία μόνο κατεύθυνση, από τους κόμβους εισόδου προς τους κόμβους εξόδου μέσω των κρυφών κόμβων. Επίσης, οι συνδέσεις μεταξύ των κόμβων δεν σχηματίζουν κύκλο, δεν υπάρχει δηλαδή ανατροφοδότηση της εξόδου ενός νευρώνα προς τους νευρώνες από τους οποίους επηρεάζεται άμεσα ή έμμεσα.

Το πιο απλό είδος νευρωνικού δικτύου που αναπτύχθηκε είναι το νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης Ενός Επιπέδου το οποίο έχει τα χαρακτηριστικά ενός δικτύου εμπρόσθιας τροφοδότησης και αποτελείται από ένα μόνο επίπεδο κόμβων εξόδου. Οι είσοδοι του δικτύου τροφοδοτούνται απευθείας στις εξόδους μέσω μιας σειράς βαρών. Το άθροισμα των βαρών και των εισόδων υπολογίζεται σε κάθε κόμβο και εάν η τιμή είναι μεγαλύτερη από κάποιο όριο, ενεργοποιείται η αντίστοιχη έξοδος.

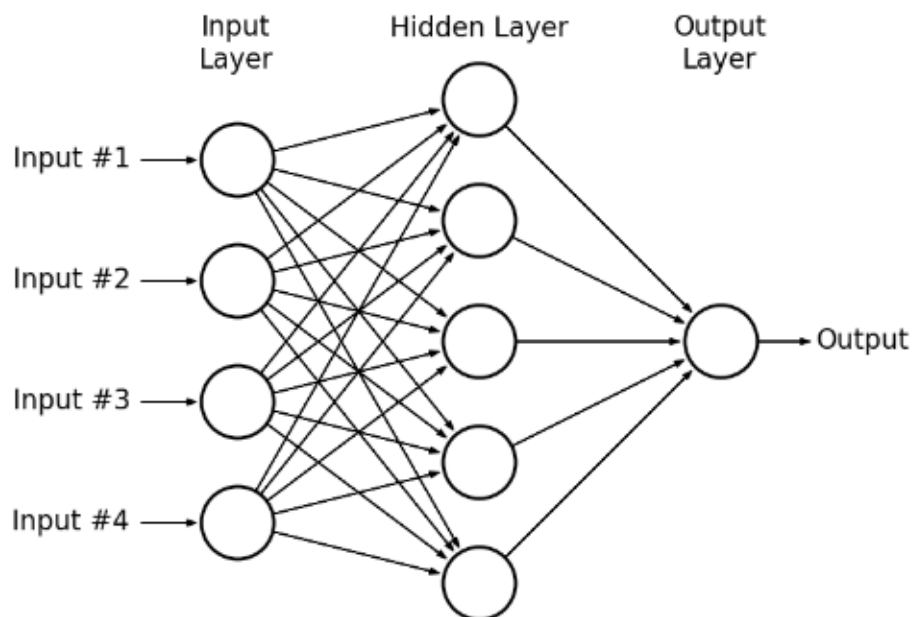
$$y = \begin{cases} 1 & \text{if } g(z) \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad (2.1)$$

Το πιο απλό νευρωνικό δίκτυο είναι το Perceptron το οποίο αποτελείται από έναν μόνο νευρώνα [7]. Εννοιολογικά, το Perceptron λειτουργεί με τρόπο παρόμοιο με έναν βιολογικό νευρώνα.

Είναι ουσιαστικά ένας δυαδικός ταξινομητής, δηλαδή μία συνάρτηση η οποία απεικονίζει την είσοδο σε μία τιμή εξόδου. Ένα Perceptron μπορεί να δημιουργηθεί εκπαιδεύοντάς το με έναν απλό αλγόριθμο εκμάθησης που υπολογίζει τα σφάλματα μεταξύ των υπολογισμένων εξόδων και των δεδομένων εξόδων για να δημιουργήσει μια προσαρμογή στα βάρη που συνδέουν τις εισόδους με την έξοδο.

### 2.2.2 Νευρωνικό Δίκτυο Εμπρόσθιας Τροφοδότησης Πολλών Επιπέδων (Multi-Layer FNN)

Σε αυτήν την ενότητα, θα αναπτυχθεί ο τρόπος σύνδεσης πολλών μεμονωμένων νευρώνων σε τουλάχιστον ένα «κρυφό στρώμα» νευρώνων μεταξύ στους κόμβους εισόδου και εξόδου. Και σε αυτήν την περίπτωση λόγω του ότι έχουμε εμπρόσθια τροφοδότηση, δεν έχουμε ανατροφοδότηση της εξόδου ενός νευρώνα προς τα «πίσω». Στο Σχήμα 2.2, φαίνεται ένα νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης τριών επιπέδων, ένα επίπεδο εισόδου, ένα κρυφό στρώμα και ένα επίπεδο εξόδου. Οι νευρώνες του κρυφού στρώματος είναι πλήρως συνδεδεμένες με το επίπεδο εισόδου και το επίπεδο εξόδου πλήρως συνδεδεμένο με το κρυφό αντίστοιχα. Σε κάθε σύνδεση μεταξύ των κόμβων εκχωρούνται τα βάρη που αντικατοπτρίζουν τη σημασία της επιρροής τους στην τελική πρόβλεψη. Εάν ένα τέτοιο δίκτυο έχει περισσότερα από ένα κρυφό στρώμα, το ονομάζουμε βαθύ τεχνητό νευρωνικό.



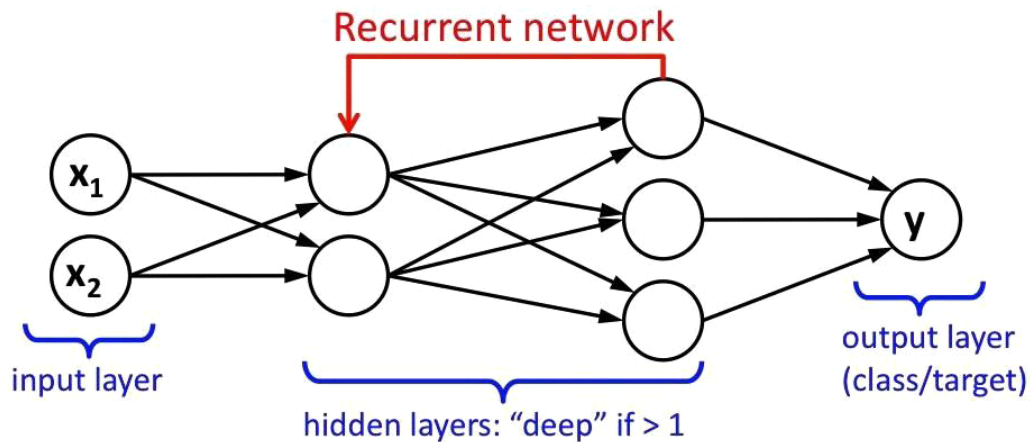
Σχήμα 2.2 Νευρωνικό Δίκτυο Εμπρόσθιας Τροφοδότησης

Ένα νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης πολλών επιπέδων μπορεί να έχει ένα ή περισσότερα κρυμμένα επίπεδα. Κάθε επίπεδο μπορεί επίσης να έχει έναν ή περισσότερους

νευρώνες [8]. Επομένως, κατά τη δημιουργία ενός τέτοιου δικτύου, ορίζεται ο αριθμός των κρυμμένων επιπέδων καθώς και ο αριθμός των νευρώνων του. Όσο πιο πολλά τα επίπεδα, δηλαδή όσο πιο βαθύ είναι το δίκτυο, τόσο περισσότερο θα προσαρμόζεται στα δεδομένα εκπαίδευσης. Ωστόσο, αυτό δεν έχει πάντα θετικά αποτελέσματα στην εκπαίδευσή του. Όταν προσαρμόζεται πολύ στα δεδομένα εκπαίδευσης τόσο πιο δύσκολα καταφέρνει να γενικευτεί σε νέα δεδομένα που του δίνονται. Αυτό ονομάζεται υπερπροσαρμογή (ting). Επίσης, τα περισσότερα επίπεδα κάνουν το δίκτυο να γίνεται υπολογιστικά ακριβότερο με αποτέλεσμα να εκπαιδεύεται πιο αργά.

### 2.2.3 Αναδρομικά Νευρωνικά Δίκτυα (Recurrent Neural Networks)

Τα αναδρομικά νευρωνικά δίκτυα ή αλλιώς επαναλαμβανόμενα νευρωνικά δίκτυα μπορούν να θεωρηθούν ως νευρωνικά δίκτυο εμπρόσθιας τροφοδότησης πολλών επιπέδων με την διαφορά ότι διαθέτουν βρόγχους ανατροφοδότησης [9].



Σχήμα 2.3 Αρχιτεκτονική Αναδρομικού Νευρωνικού Δικτύου [15]

Αυτή η αρχιτεκτονική επιτρέπει στο μοντέλο να «θυμάται» χρήσιμες πληροφορίες σχετικά με την είσοδο του δικτύου. Αυτό καθιστά τα αναδρομικά νευρωνικά δίκτυα κατάλληλα για την επεξεργασία ακολουθιών μεταβλητού μήκους εισόδων και συναντά εφαρμογές στην αναγνώριση γραφής ή ομιλίας. Τα αναδρομικά νευρωνικά δίκτυα, συνήθως χρησιμοποιούν μικρά μοντέλα καθώς όταν χρησιμοποιούνται σε νευρωνικά πολλών επιπέδων, έχει αποδειχθεί πως συμβάλλουν στην υπερπροσαρμογή (overfit).

### 2.3 ΣΥΝΑΡΤΗΣΕΙΣ ΕΝΕΡΓΟΠΟΙΗΣΗΣ (ACTIVATION FUNCTIONS)

Κατά τον σχεδιασμό ενός τεχνητού νευρωνικού δικτύου, θα πρέπει να αποφασιστεί ποια θα είναι η συνάρτηση ενεργοποίησης για τους υπολογισμούς του κάθε νευρώνα. Οι συναρτήσεις

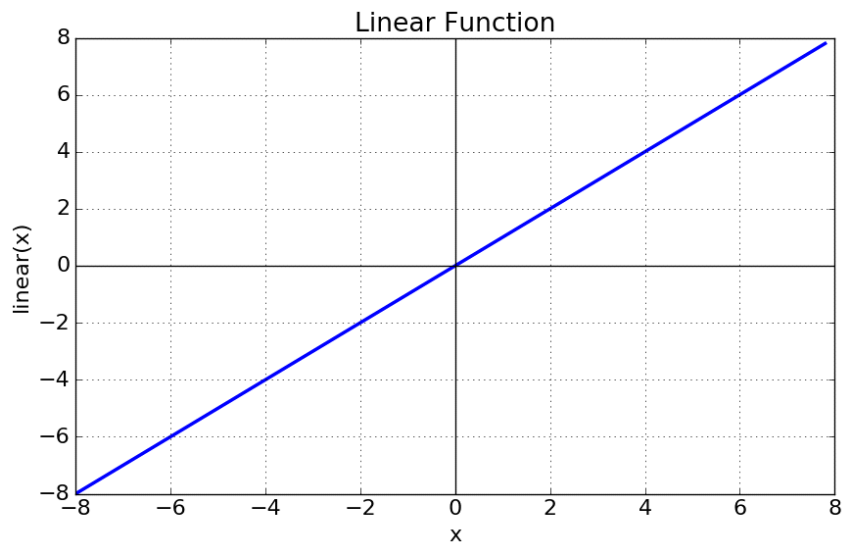
ενεργοποίησης αναφέρονται επίσης και ως συναρτήσεις μεταφοράς ή μη γραμμικότητας επειδή μετατρέπουν το γραμμικό συνδυασμό του αθροίσματος εισόδου και των βαρών σε ένα μη γραμμικό μοντέλο. Τοποθετούνται στην έξοδο κάθε νευρώνα (ή στρώματος νευρώνων) και αποφασίζει εάν θα ενεργοποιηθεί αυτός ο νευρώνας. Χωρίς αυτές, ένα νευρωνικό δίκτυο πολλών επιπέδων θα είχε την ίδια απόδοση με ένα νευρωνικό δίκτυο ενός επιπέδου. Οι συναρτήσεις ενεργοποίησης είναι απαραίτητες για τον περιορισμό της τιμής εξόδου του κάθε νευρώνα σε μία πεπερασμένη τιμή.

Υπάρχουν πάρα πολλές διαφορετικές συναρτήσεις ενεργοποίησης και με την πάροδο του χρόνου δημιουργούνται νέες. Ωστόσο λίγες από αυτές αντιστοιχούν στη συντριπτική πλειονότητα των αναγκών ενεργοποίησης. Παρακάτω θα αναλυθούν μερικές από αυτές.

### 2.3.1 Γραμμική Συνάρτηση ενεργοποίησης (Linear Function)

Μια γραμμική συνάρτηση μεταφοράς, που ονομάζεται επίσης και συνάρτηση ταυτότητας, μεταδίδει ένα σήμα αμετάβλητο. Πρακτικά, η έξοδος είναι ίση με την είσοδο, που σημαίνει ότι στην ουσία είναι σαν να μην έχουμε συνάρτηση ενεργοποίησης. Έτσι, ανεξάρτητα από το πόσα επίπεδα έχει το νευρωνικό μας δίκτυο, το μόνο που κάνει είναι να υπολογίζει μια γραμμική λειτουργία ενεργοποίησης.

$$activation(z) = w * x + b \quad (2.2)$$



Σχήμα 2.4 Γραμμική Συνάρτηση Ενεργοποίησης [17]

### 2.3.2 Βηματική Συνάρτηση Ενεργοποίησης (Binary Step)

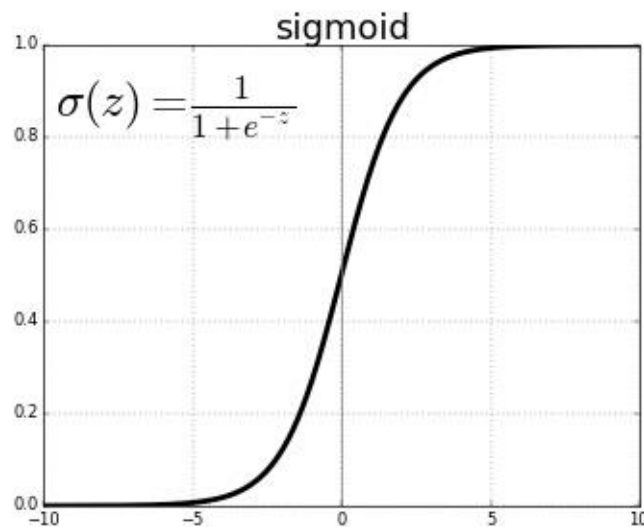
Η βηματική συνάρτηση παράγει δυαδική έξοδο και μπορεί να είναι η

$$\varphi(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.3)$$

ή οποιαδήποτε άλλη βηματική συνάρτηση. Στα τεχνητά νευρωνικά δίκτυα δεν θεωρείται χρήσιμη ως συνάρτηση ενεργοποίησης καθώς η παράγωγος της απειρίζεται.

### 2.3.3 Σιγμοειδής Συνάρτηση Ενεργοποίησης (Sigmoid Function)

Η σιγμοειδής συνάρτηση ενεργοποίησης μοιάζει με τη βηματική με την διαφορά ότι είναι συνεχής και παραγωγίσιμη σε όλο το πεδίο ορισμού της. Είναι μια από τις πιο κοινές συναρτήσεις ενεργοποίησης και χρησιμοποιείται συχνά σε δυαδικούς ταξινομητές για να προβλέπει την πιθανότητα μιας κλάσης όταν έχουμε δύο κλάσεις. Η σιγμοειδής συνάρτηση κανονικοποιεί όλες τις τιμές σε μία πιθανότητα μεταξύ μηδέν και ένα.



Σχήμα 2.5 2.3.3 Σιγμοειδής Συνάρτηση Ενεργοποίησης [18]

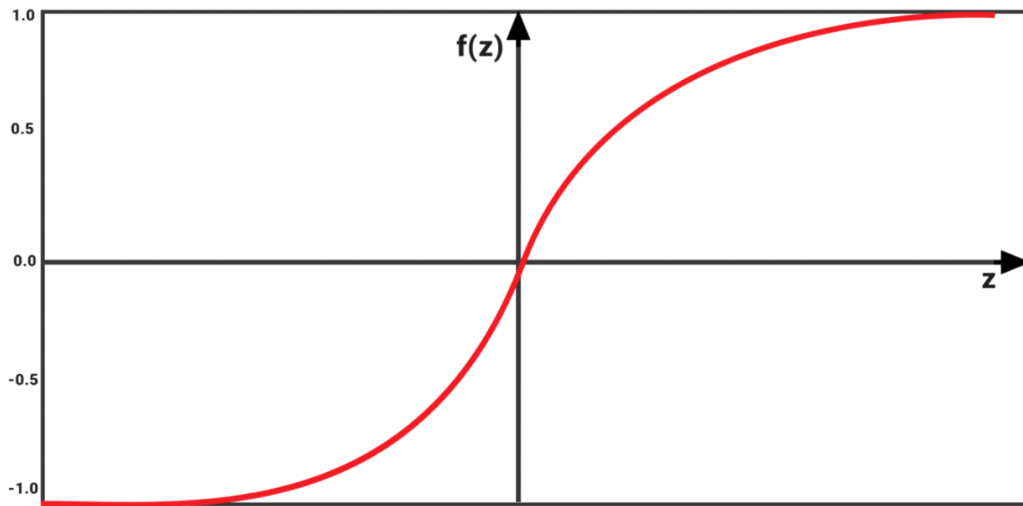
Έτσι ακόμα και όταν οι εισοδοί είναι πολύ μεγάλοι αριθμοί δεν δημιουργείται κανένα πρόβλημα. Όσο πιο αρνητική είναι μια είσοδος, τόσο πιο κοντά στο μηδέν την τοποθετεί, ενώ όσο πιο θετική, τόσο πιο κοντά στο ένα.

### 2.3.4 Συνάρτηση Υπερβολικής Εφαπτομένης (tanh Function)

Η συνάρτηση υπερβολικής εφαπτομένης είναι μια μετατοπισμένη έκδοση της σιγμοειδούς. Αντί να κανονικοποιεί όλες τις τιμές σήματος από 0 έως 1, τις κανονικοποιεί στο εύρος -1 έως 1. Η συνάρτηση υπερβολικής εφαπτομένης, λειτουργεί σχεδόν πάντα καλύτερα από την σιγμοειδή σε κρυμμένα επίπεδα καθώς έχει ως αποτέλεσμα το κεντράρισμα των δεδομένων έτσι ώστε ο

μέσος όρος των δεδομένων να είναι κοντά στο μηδέν αντί για 0,5 που είναι στη σιγμοειδή, γεγονός που καθιστά τη μάθηση για το επόμενο επίπεδο λίγο πιο εύκολη. Η εξίσωση της συνάρτησης υπερβολικής εφαπτομένης:

$$g(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (2.4)$$



Σχήμα 2.6 Συνάρτηση Υπερβολικής Εφαπτομένης

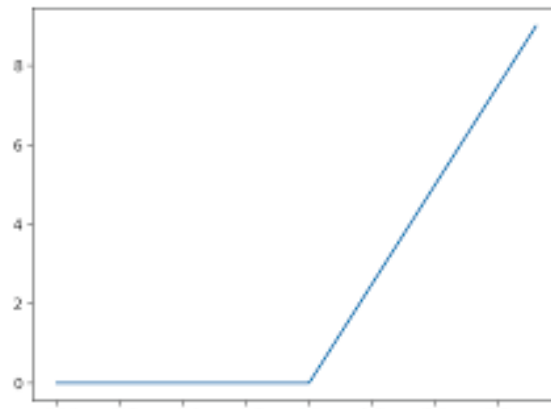
Ένα από τα μειονεκτήματα των σιγμοειδών συναρτήσεων καθώς και των συναρτήσεων υπερβολικής εφαπτομένης είναι ότι η κλίση της συνάρτησης μπορεί να γίνει πολύ μικρή.

### 2.3.5 Ανορθωμένη Γραμμική Συνάρτηση Ράμπας (Rectified Linear Unit Function)

Η ανορθωμένη γραμμική συνάρτηση ράμπας είναι η πιο συχνά χρησιμοποιούμενη συνάρτηση ενεργοποίησης στα μοντέλα βαθιάς μάθησης αυτήν τη στιγμή.

$$y = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (2.5)$$

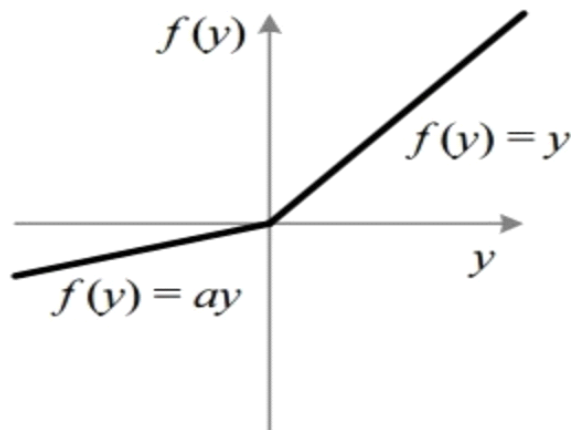
Όπως φαίνεται στο Σχήμα 2.7, η έξοδος της συνάρτησης είναι μηδέν όταν η είσοδος είναι μικρότερη από μηδέν και ίση με την τιμή της όταν είναι μεγαλύτερη από μηδέν. Η ανορθωμένη γραμμική συνάρτηση ράμπας διορθώνει κατά το ήμισυ το πρόβλημα των σιγμοειδών συναρτήσεων και των συναρτήσεων υπερβολικής εφαιπτομένης αλλά το ζήτημα είναι πως όλες οι αρνητικές τιμές γίνονται μηδέν, γεγονός που μειώνει την ικανότητα του μοντέλου να εκπαιδεύεται σωστά από αυτά τα δεδομένα.



Σχήμα 2.7 Ανορθωμένη Γραμμική Συνάρτηση Ράμπας

### 2.3.6 Διακεκομμένη Ανορθωμένη Γραμμική Συνάρτηση Ράμπας (leaky-ReLU)

Η διακεκομμένη ανορθωμένη γραμμική συνάρτηση ράμπας είναι μια τροποποίηση της ανορθωμένης γραμμικής συνάρτησης ράμπας [10]. Αυτή η συνάρτηση, αντιμετωπίζει το πρόβλημα της ανορθωμένης γραμμικής συνάρτησης ράμπας καθώς επιστρέφει τιμές και στις περιπτώσεις που η είσοδος είναι αρνητική.



Σχήμα 2.8 Διακεκομμένη Ανορθωμένη Γραμμική Συνάρτηση Ράμπας



Όπως φαίνεται και στο Σχήμα 2.8, όταν η τιμές είναι θετικές τις επιστρέφει αυτούσιες ενώ όταν είναι αρνητικές τις πολλαπλασιάζει με έναν αριθμό. Ο αριθμός αυτός στη διακεκομμένη ανορθωμένη γραμμική συνάρτηση ράμπας είναι το 0.01. Έτσι το μοντέλο μπορεί να εκπαιδεύεται και στις αρνητικές τιμές.

### 2.3.7 Συνάρτηση Ενεργοποίησης Softmax

Η συνάρτηση ενεργοποίησης Softmax είναι μια πιο γενικευμένη συνάρτηση που χρησιμοποιείται για την ταξινόμηση πολλαπλών κλάσεων. Όταν έχουμε δηλαδή περισσότερες από δύο εξόδους, προσδιορίζει την πιθανότητα κατανομής για καθεμία από αυτές. Χρησιμοποιείται γενικά στο τελευταίο επίπεδο νευρώνων για την επίλυση ενός προβλήματος ταξινόμησης.

$$\sigma(x) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}} \quad (2.6)$$

## 2.4 ΣΥΝΑΡΤΗΣΕΙΣ ΣΦΑΛΜΑΤΟΣ (LOSS FUNCTIONS)

Στόχος ενός τεχνητού νευρωνικού δικτύου είναι να βελτιώνεται με την πάροδο του χρόνου. Κατά την εκπαίδευση του δικτύου τροποποιούνται τα βάρη που συνδέουν τους νευρώνες του ώστε να αυξηθεί η ακρίβεια των προβλέψεών του. Για να γίνει αυτό, υπολογίζεται πόσο μεγάλο είναι το σφάλμα το προβλέψεων του δικτύου σε σχέση με την αναμενόμενη έξοδο της κάθε εισόδου. Η συνάρτηση απώλειας ή αλλιώς συνάρτηση κόστους, είναι ο αλγόριθμος που ποσοτικοποιεί αυτό το σφάλμα [11]. Έτσι, στην εκπαίδευση του μοντέλου μας, σκοπός μας είναι να βρούμε τις βέλτιστες μεταβλητές (βάρη) που ελαχιστοποιούν το σφάλμα όσο το δυνατό περισσότερο. Δεδομένου ότι η απώλεια είναι το σφάλμα του μοντέλου μας, θέλουμε ιδανικά να είναι μηδέν [12].

### 2.4.1 Μέσο Τετραγωνικό Σφάλμα (Mean Squared Error)

Το μέσο τετραγωνικό σφάλμα υπολογίζεται ως ο μέσος όρος της διαφορών των τετραγώνων μεταξύ των προβλεπόμενων και των αναμενόμενων τιμών.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (2.7)$$

Το αποτέλεσμα είναι πάντα θετικό και η ιδανική τιμή είναι το μηδέν. Η συνάρτηση του μέσου τετραγωνικού σφάλματος χρησιμοποιείται για τη λύση προβλημάτων παλινδρόμησης.

#### 2.4.2 Συνάρτηση Σφάλματος Binary Cross-Entropy

Η συνάρτηση σφάλματος Binary Cross-Entropy χρησιμοποιείται για προβλήματα δυαδικής ταξινόμησης. Δηλαδή σε περιπτώσεις όπου ο κόμβος στην έξοδο του νευρωνικού δικτύου ταξινομεί τα δεδομένα σε δύο κλάσεις. Κάθε προβλεπόμενη πιθανότητα συγκρίνεται με την πραγματική τιμή εξόδου της κλάσης και υπολογίζεται μια βαθμολογία που τιμωρεί την πιθανότητα με βάση την απόσταση από την αναμενόμενη τιμή. Η συνάρτηση σφάλματος είναι λογαριθμική, που σημαίνει ότι η τιμή που επιστρέφει είναι πολύ μικρή για μικρές διαφορές και τεράστια για μεγάλες διαφορές.

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N y_i \log(p(y_i)) + (1 - y_i) * \log(1 - (p(y_i))) \quad (2.8)$$

Η συνάρτησης σφάλματος Binary Cross-Entropy υπολογίζεται ως το μέσο σφάλμα σε όλες τις τιμές εισόδου. Όπως είναι λογικό, οι μικρότερες τιμές της συνάρτησης σφάλματος Binary Cross-Entropy αντιπροσωπεύουν ένα καλύτερο μοντέλο απ' ότι οι μεγάλες. Ένα μοντέλο που προβλέπει τέλεια τις εξόδους έχει σφάλμα ίσο με μηδέν.

#### 2.4.3 Συνάρτηση Σφάλματος Categorical Cross-entropy

Η συνάρτηση σφάλματος Categorical Cross Entropy είναι παρόμοια με τη συνάρτηση σφάλματος Binary Cross-Entropy αλλά χρησιμοποιείται για προβλήματα ταξινόμησης πολλών κλάσεων. Δουλεύει ακριβώς με τον ίδιο τρόπο με τη Binary Cross-Entropy με τη διαφορά ότι δίνει στην αναμενόμενη κλάση εξόδου την τιμή ένα και σε όλες τις υπόλοιπες μηδέν. Η μαθηματική εξίσωση είναι:

$$L(y, \tilde{y}) = - \sum_{j=0}^M \sum_{i=0}^N (y_{ij} \log(\tilde{y}_i)) \quad (2.9)$$

#### 2.4.4 Συνάρτηση Σφάλματος Poisson

Η συνάρτηση απώλειας Poisson χρησιμοποιείται για την επίλυση προβλημάτων παλινδρόμησης με αριθμητικά δεδομένα. Η συνάρτηση απώλειας Poisson είναι μια παραλλαγή της κατανομής

Poisson. Ένα παράδειγμα χρήσης αυτής της συνάρτησης είναι η εύρεση του αριθμού των πελατών που θα εισέλθουν σε ένα κατάστημα. Η μαθηματική εξίσωση είναι:

$$L(y, \tilde{y}) = \frac{1}{N} \sum_{i=0}^n (\tilde{y}_i - y_i \log \tilde{y}_i) \quad (2.10)$$

## 2.5 ΑΛΓΟΡΙΘΜΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ (OPTIMIZATION ALGORITHMS)

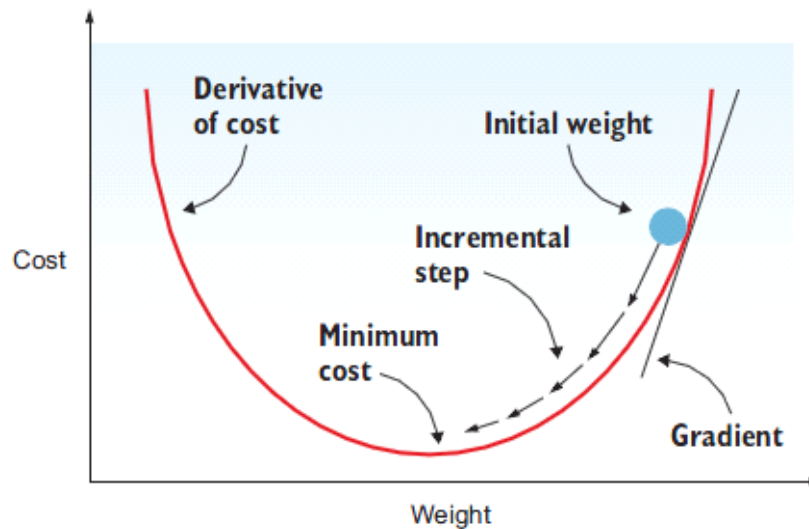
Όπως έχει προαναφερθεί, η εκπαίδευση ενός νευρωνικού δικτύου περιλαμβάνει πολλά ζεύγη δεδομένων (οι εισόδοι και τις ετικέτες τους). Το δίκτυο κάνει προβλέψεις για τις εισόδους και τις συγκρίνει με τις ετικέτες τους προκειμένου να υπολογίσει το σφάλμα. Τέλος, το νευρωνικό δίκτυο προσαρμόζει τα βάρη μέχρι να ελαχιστοποιήσει την τιμή του σφάλματος. Οι αλγόριθμοι που βρίσκουν τα βέλτιστα βάρη κάθε φορά ονομάζονται αλγόριθμοι βελτιστοποίησης.

Το γεγονός ότι υπολογίζουμε το σφάλμα κατά την εκπαίδευση ενός νευρωνικού δικτύου μετατρέπει το πρόβλημα των νευρωνικών δικτύων σε ένα πρόβλημα βελτιστοποίησης όπου ο στόχος είναι να ελαχιστοποιηθεί το σφάλμα. Θα μπορούσαμε απλώς να αλλάζουμε τα βάρη χρησιμοποιώντας τυχαίες τιμές και στο τέλος να κρατούσαμε αυτές που δίνουν το μικρότερο σφάλμα. Αυτό θα ήταν μια λύση σε μικρά νευρωνικά δίκτυα με λίγες παραμέτρους. Ωστόσο, προφανώς αυτή η λύση δεν θα είχε τα βέλτιστα αποτελέσματα σε κάθε περίπτωση καθώς επίσης θα ήταν πρακτικά αδύνατο για μεγαλύτερα και πιο περίπλοκα νευρωνικά δίκτυα, που συνήθως χρησιμοποιούνται σε πρακτικές εφαρμογές. Παρακάτω παρατίθενται οι σημαντικότεροι αλγόριθμοι βελτιστοποίησης.

### 2.5.1 Αλγόριθμος Ομαδικής Κατάβασης Κλίσης (Batch Gradient Descent)

Ο αλγόριθμος κατάβασης κλίσης είναι ένας από τους πιο διάσημους αλγορίθμους βελτιστοποίησης που χρησιμοποιείται κυρίως σε προβλήματα γραμμικής παλινδρόμησης και σε αλγορίθμους ταξινόμησης [12]. Βασίζεται σε μια κυρτή συνάρτηση και τροποποιεί τις παραμέτρους τις επαναληπτικά για να ελαχιστοποιήσει μια δεδομένη συνάρτηση στο τοπικό της ελάχιστο.

Αρχικά καθορίζονται οι τιμές των αρχικών βαρών (initial weight), όπως φαίνεται και στο Σχήμα 2.9 και από εκεί προσαρμόζει τα βάρη επαναληπτικά μειώνοντας την κλίση.



Σχήμα 2.9 Αλγόριθμος Ομαδικής Κατάβασης Κλίσης

Το πόσο μεγάλη είναι η αλλαγή της κλίσης που εφαρμόζεται κάθε φορά προς την κατεύθυνση του τοπικού ελαχίστου καθορίζεται από το ποσοστό εκμάθησης (learning rate). Για να φτάσει η κλίση στο τοπικό ελάχιστο πρέπει να ορίσουμε το ποσοστό εκμάθησης στην κατάλληλη τιμή, η οποία δεν είναι ούτε πολύ μεγάλη ούτε πολύ μικρή. Εάν το ποσοστό εκμάθησης είναι πολύ μεγάλο, ενδέχεται να μην φτάσουμε ποτέ στο τοπικό ελάχιστο επειδή θα αναπηδά μεταξύ των κυρτών συντελεστών της κλίσης. Στην περίπτωση τώρα που το ποσοστό εκμάθησης είναι πολύ μικρό, η μετάβαση της κλίσης θα φτάσει τελικά στο τοπικό ελάχιστο αλλά μετά από αρκετή ώρα.

Η ομαδική κατάβαση κλίσης υπολογίζει το σφάλμα στο σύνολο των δεδομένων εκπαίδευσης και αφού αξιολογηθούν όλα τα ζευγάρια, το μοντέλο ενημερώνει τα βάρη του. Αυτή η διαδικασία μοιάζει με έναν κύκλο και ονομάζεται εποχική εκπαίδευση. Έτσι, το μοντέλο παράγει μια σταθερή διαβάθμιση σφάλματος και μια σταθερή σύγκλιση. Ωστόσο, μπορεί μερικές φορές να οδηγήσει σε μια κατάσταση σύγκλισης που δεν είναι η καλύτερη που μπορεί να επιτύχει το μοντέλο. Ένα άλλο μειονέκτημα είναι ότι απαιτείται ολόκληρο το σύνολο δεδομένων να είναι αποθηκευμένο στη μνήμη και να είναι διαθέσιμο στον αλγόριθμο με αποτέλεσμα να μην μπορεί να χρησιμοποιηθεί για μεγάλα σετ δεδομένων.

### 2.5.2 Στοχαστικός Αλγόριθμος Κατάβασης Κλίσης (Stochastic Gradient Descent)

Ο στοχαστικός αλγόριθμος κατάβασης κλίσης είναι μια παραλλαγή του αλγορίθμου απότομης καθόδου. Η διαφορά τους είναι ότι ο στοχαστικός αλγόριθμος κατάβασης κλίσης ενημερώνει τα βάρη για κάθε ζεύγος δεδομένων ένα προς ένα αντί να υπολογίζει το σφάλμα στο σύνολο των δεδομένων εκπαίδευσης. Ένα πλεονέκτημα αυτού είναι ότι οι συχνές ενημερώσεις μας επιτρέπουν να έχουμε έναν αρκετά λεπτομερή ρυθμό βελτίωσης. Ωστόσο, οι συχνές ενημερώσεις είναι υπολογιστικά πιο ακριβές και μπορεί να οδηγήσουν σε άλματα του ποσοστού του σφάλματος αντί σε σταδιακή μείωση.

### 2.5.3 Αλγόριθμος Κατάβασης Κλίσης Μικρής Παρτίδας (Mini-Batch Gradient Descent)

Ο αλγόριθμος κατάβασης κλίσης μικρής παρτίδας είναι ο πιο αποδοτικός αλγόριθμος κατάβασης κλίσης. Συνδυάζει τα πλεονεκτήματα και των δύο προηγούμενων αλγορίθμων. Χωρίζει το σύνολο δεδομένων εκπαίδευσης σε μικρές παρτίδες και ενημερώνει τα βάρη σε καθεμία από αυτές. Δεν υπάρχει κάποιος σαφής κανόνας για το μέγεθος αυτής της παρτίδας αλλά ποικίλει για διαφορετικές εφαρμογές. Αυτός είναι ο πιο συνηθισμένος αλγόριθμος βελτιστοποίησης στη βαθιά μάθηση.

### 2.5.4 Αλγόριθμος Βελτιστοποίησης Adam

Ο αλγόριθμος βελτιστοποίησης Adam είναι μια επέκταση του στοχαστικού αλγορίθμου κατάβασης κλίσης και μπορεί να χρησιμοποιηθεί για την πιο αποτελεσματική ενημέρωση του δικτύου. Ο αλγόριθμος βελτιστοποίησης Adam χρησιμοποιεί τον αλγόριθμο momentum και τον αλγόριθμο προσαρμογής του ποσοστού μάθησης.

Ο αλγόριθμος momentum μπορεί να προσομοιωθεί με μια μπάλα που κυλά προς τα τοπικά ελάχιστα. Ουσιαστικά, κάθε φορά που ανανεώνονται τα βάρη, το ποσοστό εκμάθησης μεγαλώνει. Αυτό έχει ως αποτέλεσμα την σταδιακή αύξηση του ποσοστού εκμάθησης. Η μαθηματική εξίσωση είναι:

$$Vt = \gamma Vt - 1 + n \nabla \theta, (\theta) \quad (2.11)$$

$$\theta = \theta - Vt$$

Στην παραπάνω εξίσωση, η παράμετρος  $\gamma$  παίρνει συνήθως την τιμή 0,9.

Η κεντρική ιδέα του αλγορίθμου προσαρμογής του ποσοστού μάθησης είναι να «θυμάται» τον κινούμενο μέσο όρο των τετραγωνικών κλίσεων για κάθε βάρος. Έπειτα, διαιρεί τη κλίση με τη τετραγωνική ρίζα του μέσου όρου τους. Η μαθηματική απεικόνιση είναι η παρακάτω:

$$E[g^2] = \gamma E[g^2]_{t-1} + (1 - \gamma) \frac{\delta C}{\delta w} \quad (2.12)$$

$$\vartheta_{t+1} = \theta_t - \frac{n}{\sqrt{E[g^2]_t}} * \left( \frac{\delta C}{\delta w} \right)^2, \text{ όπου } \left( \frac{\delta C}{\delta w} \right) = g_t$$

Και εδώ, η παράμετρος  $\gamma$  λαμβάνει την τιμή 0,9 και μια προεπιλεγμένη καλή τιμή του ρυθμού μάθησης  $\eta$  είναι 0.001.

Ο αλγόριθμος βελτιστοποίησης Adam χρησιμοποιώντας τους δύο αυτούς αλγορίθμους συγκλίνει υπερβολικά γρήγορα και γι' αυτό συγκριτικά με όλου τους άλλους αλγόριθμους βελτιστοποίησης είναι ο πιο αποδοτικός καθώς τα τεχνητά νευρωνικά δίκτυα εκπαιδεύονται σε λιγότερο χρονικό διάστημα με μεγαλύτερη ακρίβεια.

## 2.6 ΑΛΓΟΡΙΘΜΟΣ ΟΠΙΣΘΟΔΙΑΔΟΣΗΣ (BACKPROPAGATION ALGORITHM)

Ο αλγόριθμος οπισθοδιάδοσης είναι ο πυρήνας του πως μαθαίνουν τα τεχνητά νευρωνικά δίκτυα. Όπως έχουμε δει ως τώρα, η εκπαίδευση ενός νευρωνικού δικτύου γίνεται με την επανάληψη των παρακάτω τριών βημάτων.

Το σήμα μεταφέρεται προς τα εμπρός μέσα από τα κρυφά επίπεδα έως ότου φτάσει στο επίπεδο εξόδου εφαρμόζοντας τις συναρτήσεις ενεργοποίησης προκειμένου να ληφθεί μια πρόβλεψη για την έξοδο.

Γίνεται σύγκριση της πρόβλεψης με την αναμενόμενη έξοδο και υπολογίζεται το σφάλμα μέσω της συνάρτησης σφάλματος.

Με τη χρήση του αλγορίθμου βελτιστοποίησης μεταβάλλονται τα βάρη που συνδέουν τους νευρώνες μεταξύ τους έτσι ώστε να υπάρχει μείωση του σφάλματος.

Ο αλγόριθμος οπισθοδιάδοσης στοχεύει στην ελαχιστοποίηση του σφάλματος προσαρμόζοντας τα βάρη του δικτύου από το τελευταίο επίπεδο νευρώνων (έξοδος) πίσω στο πρώτο (είσοδος) λαμβάνοντας υπόψη τα ήδη υπάρχοντα. Τα βάρη αλλάζουν σύμφωνα με την παρακάτω μαθηματική εξίσωση.

$$W_{next-step} = W_{current} + \Delta W \quad (2.13)$$

Όπου  $W_{next-step}$  είναι τα νέα βάρη,  $W_{current}$  τα προηγούμενα και  $\Delta w$  το ποσοστό σφάλματος στην προηγούμενη εποχή. Τα πλεονεκτήματα του αλγορίθμου οπισθοδιάδοσης είναι ότι είναι εύκολο και γρήγορο να προγραμματιστεί, πέρα από τον αριθμό των εισόδων δεν έχει άλλες παραμέτρους που πρέπει να αρχικοποιηθούν, είναι μια ευέλικτη μέθοδος εκπαίδευσης

νευρωνικών δικτύων καθώς δεν απαιτεί προηγούμενη γνώση σχετικά με το δίκτυο και είναι μια τυπική μέθοδος που λειτουργεί γενικά καλά.

## ΚΕΦΑΛΑΙΟ 3ο

---

### Συνελικτικά Νευρωνικά Δίκτυα (CNN)

#### 3.1 ΕΙΣΑΓΩΓΗ

Παρόλο που εδώ και πολλά χρόνια έχουν αναπτυχθεί πολύ ισχυροί αλγόριθμοι τεχνητών νευρωνικών δικτύων όπως για παράδειγμα μηχανές που κερδίζουν τους παγκόσμιους πρωταθλητές στο σκάκι, μόλις πρόσφατα οι υπολογιστές μπόρεσαν να εξάγουν πληροφορίες από μια εικόνα. Τα συνελικτικά νευρωνικά δίκτυα προέκυψαν από τη μελέτη της οπτικής του εγκεφάλου και έχουν χρησιμοποιηθεί στην αναγνώριση εικόνας από το 1980 [7]. Τα τελευταία χρόνια, κυρίως χάρη στην αύξηση της υπολογιστικής ισχύος και στην ανάπτυξη νέων τεχνικών για την εκπαίδευση συνελικτικών νευρωνικών δικτύων, οι υπολογιστές κατάφεραν να πετύχουν πολύ υψηλές επιδόσεις σε πολύπλοκες οπτικές εργασίες. Σε αυτό συντέλεσε επίσης και η ραγδαία αύξηση του αριθμού των διαθέσιμων δεδομένων για την εκπαίδευση αυτών των δικτύων.

Τα συνελικτικά νευρωνικά δίκτυα, όπως ακριβώς και τα παλαιότερα νευρωνικά δίκτυα, περιλαμβάνουν κρυφά στρώματα νευρώνων τα οποία ενώνονται μεταξύ τους, με την είσοδο και την έξοδο του δικτύου μέσω βαρών. Τα βάρη αυτά ξεκινούν με τυχαίες τιμές και εφαρμόζοντας τις συναρτήσεις ενεργοποίησης εξάγονται προβλέψεις για την έξοδο και αφού υπολογιστεί το σφάλμα μέσω των συναρτήσεων σφάλματος ενημερώνονται τα βάρη με την διαδικασία της οπισθοδιάδοσης. Η μόνη διαφορά είναι ότι χρησιμοποιούνται συνελικτικά νευρωνικά στρώματα αντί για κανονικά.

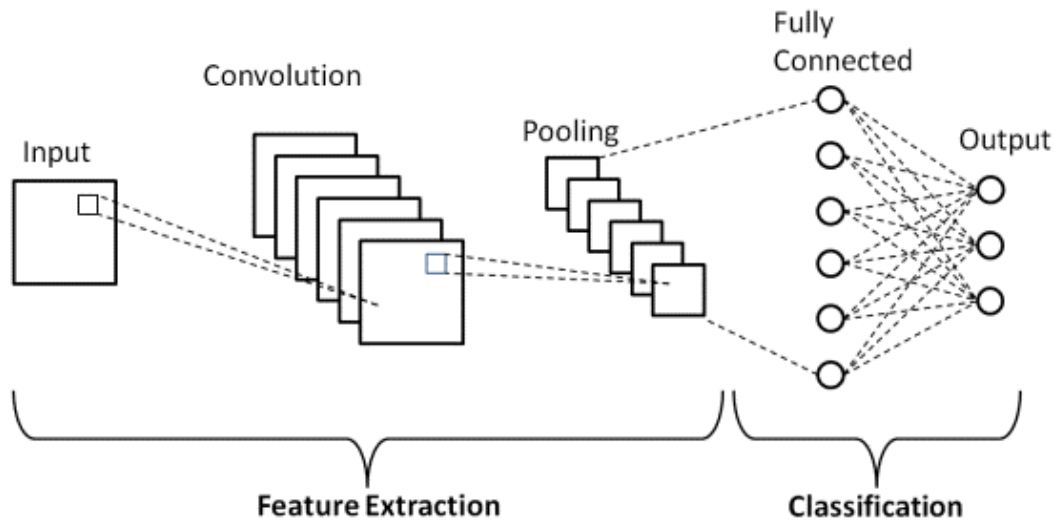
Χωρίς τη βαθιά μάθηση και τα συνελικτικά νευρωνικά δίκτυα, για την αναγνώριση ενός ανθρώπου σε μια εικόνα για παράδειγμα, έπρεπε να γίνει η εξαγωγή των χαρακτηριστικών των εικόνων εκπαίδευσης και στη συνέχεια να τροφοδοτηθούν αυτά σε ένα νευρωνικό δίκτυο ταξινόμησης. Όλη αυτή η διαδικασία μπορεί να παραληφθεί με την χρήση των συνελικτικών νευρώνων. Στα συνελικτικά νευρωνικά δίκτυα, το πρώτο επίπεδο κρυφών νευρώνων μαθαίνει μερικά βασικά χαρακτηριστικά (ακμές και γραμμές), το επόμενο επίπεδο μαθαίνει χαρακτηριστικά που είναι λίγο πιο περίπλοκα (κύκλοι, τετράγωνα), το αμέσως επόμενο βρίσκει ακόμα πιο περίπλοκα (ρόδα αυτοκινήτου, μάτια) και ούτω καθεξής.



### 3.2 ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΤΩΝ ΣΥΝΕΛΙΚΤΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ

Τα συνελκτικά νευρωνικά δίκτυα παρ' όλες τις διαφορές που μπορεί να έχουν μεταξύ τους, τα βασικά τους στοιχεία είναι τρία:

- Συνελκτικά επίπεδα νευρώνων (Convolutional layers)
- Δειγματοληπτικά επιπέδων (Pooling layers)
- Πλήρως συνδεδεμένα επίπεδα (Fully connected layers)



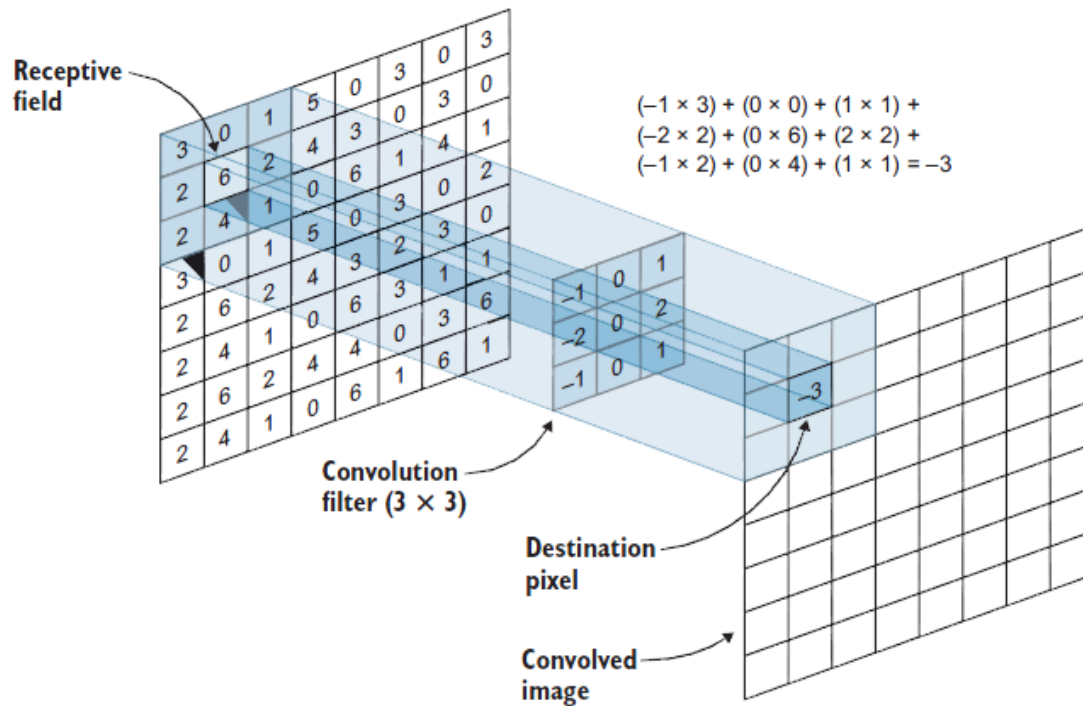
Σχήμα 3.1 Δομή Συνελκτικών νευρωνικών δικτύων [22]

Όπως φαίνεται στο Σχήμα 3.1 τα πρώτα δύο στρώματα χρησιμοποιούνται για την εξαγωγή χαρακτηριστικών από την εικόνα τα οποία στη συνέχεια θα τροφοδοτηθούν στα πλήρως συνδεδεμένα στρώματα τα οποία χρησιμοποιούνται για την ταξινόμηση

#### 3.2.1 Συνελκτικά Επίπεδα Νευρώνων (Convolutional Layers)

Στα μαθηματικά η συνέλιξη είναι μια πράξη που εφαρμόζεται σε δύο συναρτήσεις για την παραγωγή μιας τρίτης. Στα συνελκτικά νευρωνικά δίκτυα, η πρώτη συνάρτηση είναι η εικόνα εισόδου και η δεύτερη είναι το συνελκτικό φίλτρο [14].

Στην παραπάνω εικόνα φαίνεται στα αριστερά η αναπαράσταση μιας ασπρόμαυρης εικόνας με



Σχήμα 3.2 Εφαρμογή Συνελικτικού Φίλτρου Σε Ασπρόμαυρη Εικόνα

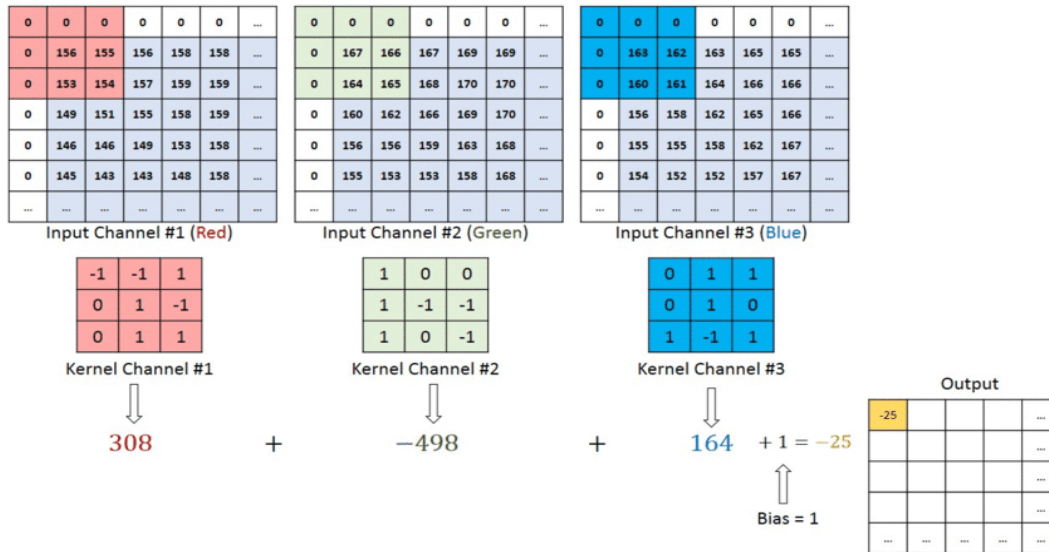
αριθμούς, στη μέση είναι ένα  $3 \times 3$  συνελικτικό φίλτρο που αλλιώς ονομάζεται kernel και στα δεξιά η παράγωγη εικόνα. Το φίλτρο kernel ολισθαίνει πάνω στην αρχική εικόνα και εφαρμόζεται σε όλα τα εικονοστοιχεία (pixel) ξεχωριστά. Η περιοχή της εικόνας που αναλύει το φίλτρο για κάθε εικονοστοιχείο ονομάζεται δεκτικό πεδίο (receptive field) Το αποτέλεσμα της συνέλιξης αποθηκεύεται στην τελική εικόνα.

Οι τιμές του φίλτρου συνέλιξης στα συνελικτικά νευρωνικά δίκτυα αποτελούν τα βάρη του. Επομένως, δεν τις ορίζουμε εμείς. Λαμβάνουν αρχικά κάποιες τυχαίες τιμές και κατά την εκπαίδευση του μοντέλου αυτές αλλάζουν με στόχο την επίτευξη προβλέψεων πιο κοντά στην αναμενόμενη έξοδο.

Ο τρόπος με τον οποίο υπολογίζεται η τιμή του κάθε εικονοστοιχείου στην τελική εικόνα είναι απλός και φαίνεται στην Σχήμα 3.2. Πολλαπλασιάζεται δηλαδή κάθε εικονοστοιχείο του δεκτικού πεδίου της αρχικής εικόνας με το αντίστοιχο του φίλτρου συνέλιξης και το άθροισμα όλων αυτών μας δίνει την τιμή του κεντρικού εικονοστοιχείου της τελικής εικόνας.

Σε περίπτωση που έχουμε έγχρωμη εικόνα, δηλαδή η αναπαράσταση της εικόνας γίνεται με περισσότερα από έναν πίνακα εικονοστοιχείων, το φίλτρο συνέλιξης έχει το ίδιο βάθος με αυτό

της εικόνας εισόδου. Αν για παράδειγμα έχουμε μια εικόνα με αναπαράσταση RGB, άρα τριών καναλιών, το φίλτρο συνέλιξης θα έχει επίσης βάθος τρία. Εδώ, το κάθε κανάλι του φίλτρου εφαρμόζεται στο αντίστοιχο της εικόνας εισόδου και στη συνέχεια αθροίζεται με τα άλλα δύο όπως φαίνεται και στο Σχήμα 3.3.



Σχήμα 3.3 Εφαρμογή Συνελικτικού Φίλτρου Σε Έγχρωμη Εικόνα

Το bias στην παραπάνω εικόνα είναι άλλη μια υπερπαραμέτρος πέρα από τα βάρη που προσδιορίζει το νευρωνικό δίκτυο κατά την διάρκεια της εκπαίδευσης με στόχο την επίτευξη ακριβέστερων προβλέψεων. Το bias, ουσιαστικά προστίθεται στο τελικό άθροισμα προτού αυτό αποθηκευτεί στην έξοδο.

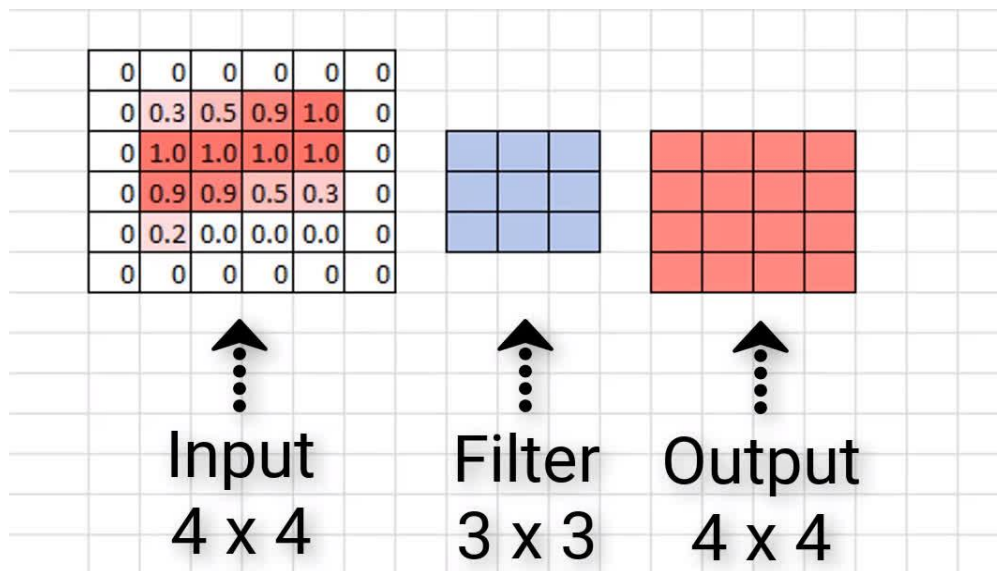
Το κάθε συνελικτικό φίλτρο στα συνελικτικά νευρωνικά δίκτυα είναι όπως ακριβώς ένας νευρώνας στα απλά νευρωνικά δίκτυα. Αυτό σημαίνει πως κάθε συνελικτικό στρώμα μπορεί να αποτελείται από περισσότερα από ένα συνελικτικά φίλτρα. Έτσι, αυξάνοντας τον αριθμό των συνελικτικών φίλτρων σε κάθε στρώμα, αυξάνουμε τον αριθμό των κρυφών μονάδων του δικτύου, γεγονός που καθιστά το δίκτυό μας πιο περίπλοκο και ικανό να ανιχνεύσει πιο περίπλοκα μοτίβα σε μια εικόνα.

Όπως έχει αναφερθεί, το συνελικτικό φίλτρο είναι ένας πίνακας βαρών που ολισθαίνει πάνω από την εικόνα προκειμένου να εξάγει κάποια χαρακτηριστικά της. Ο πίνακας αυτός είναι σχεδόν πάντα τετράγωνος και κυμαίνεται από  $2 \times 2$  έως  $5 \times 5$  και θα πρέπει να οριστεί από την αρχή. Όπως και για τις περισσότερες υπερπαραμέτρους των νευρωνικών δικτύων, δεν υπάρχει ένα στάνταρ μέγεθος που να ταιριάζει σε όλα τα προβλήματα. Θεωρητικά, όσο μεγαλύτερο είναι το

μέγεθος του πίνακα τόσο πιο βαθύ το νευρωνικό δίκτυο που σημαίνει ότι μαθαίνει καλύτερα. Όμως, όπως ακριβώς και στα απλά νευρωνικά δίκτυα, αυτό σημαίνει μεγαλύτερη υπολογιστική πολυπλοκότητα και μπορεί επίσης να οδηγήσει σε υπερπροσαρμογή (overfit). Επίσης, η χρήση μεγάλων συνελκτικών φίλτρων, αποφεύγεται διότι οδηγεί σε απώλεια σημαντικών λεπτομερειών τις εικόνας.

Δύο ακόμα πολύ βασικές υπερπαράμετροι για την εκπαίδευση ενός συνελκτικού νευρωνικού δικτύου είναι τα βήματα (Strides) και η επένδυση (Padding). Τα βήματα είναι το ποσό με το οποίο το φίλτρο ολισθαίνει πάνω από την εικόνα. Για παράδειγμα, όταν το φίλτρο μετακινείται ένα εικονοστοιχείο την φορά, η τιμή των βημάτων είναι ένα. Μπορεί να χρησιμοποιηθεί οποιοσδήποτε αριθμός βημάτων ωστόσο αποφεύγεται η χρήση τιμών μεγαλύτερων του δύο. Σε περίπτωση βήματος ένα, η εικόνα εξόδου θα έχει περίπου ίδιο ύψος και πλάτος με την εικόνα εξόδου. Σε περίπτωση βήματος δύο η εικόνα εξόδου θα είναι περίπου το μισό από την εικόνα εισόδου.

Η επένδυση (padding) συχνά αναφέρεται ως μηδενική επένδυση καθώς προσθέτουμε μηδενικά γύρω από το όριο μιας εικόνας. Η επένδυση χρησιμοποιείται για να μας επιτρέψει να εφαρμόζουμε τα φίλτρα σε όλα τα εικονοστοιχεία της εικόνας.



Σχήμα 3.4 Μηδενική επένδυση (Padding) [24]

Όπως φαίνεται στο Σχήμα 3.4 θέλουμε να εφαρμόσουμε ένα  $3 \times 3$  φίλτρο σε μια εικόνα με διαστάσεις  $4 \times 4$ . Αφού προστεθούν τα μηδενικά γύρω από την εικόνα, το φίλτρο περνά από το κάθε εικονοστοιχείο ξεχωριστά και δίνει την αντίστοιχη έξοδο. Παρατηρούμε ότι με τη βοήθεια

αυτής της μεθόδου, το πλάτος και το ύψος της εικόνας εξόδου είναι το ίδιο με της εικόνας εισόδου. Σε περίπτωση που το φίλτρο μας ήταν  $4 \times 4$ , θα έπρεπε να προστεθούν δύο στρώματα μηδενικών γύρω από την εικόνα προκειμένου να επιτευχθεί αυτό.

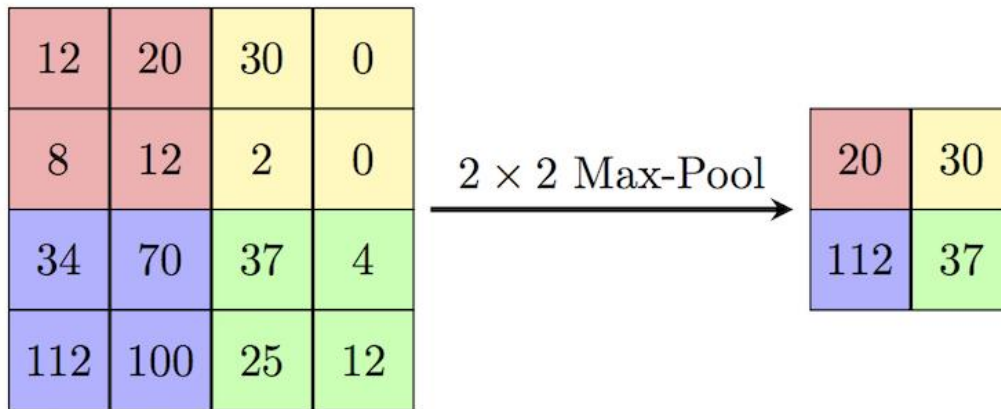
### 3.2.2 Δειγματοληπτικά Επίπεδα Ή Επίπεδα Ομαδοποίησης (Pooling Layers Or Sampling)

Τα συνελικτικά επίπεδα σε ένα συνελικτικό νευρωνικό δίκτυο δημιουργούν «χάρτες» χαρακτηριστικών που συνοψίζουν την παρουσία των χαρακτηριστικών (γραμμές, κύκλοι, πρόσωπο) στην είσοδο εφαρμόζοντας συνελικτικά φίλτρα. Ένας περιορισμός της εξόδου αυτού του χάρτη είναι ότι καταγράφουν την ακριβή θέση των χαρακτηριστικών. Αυτό έχει ως αποτέλεσμα μικρές κινήσεις στη θέση του χαρακτηριστικού στην εικόνα εισόδου που μπορεί να συμβούν κατά την επεξεργασία της, οδηγούν σε έναν νέο διαφορετικό χάρτη. Η αντιμετώπιση του προβλήματος αυτού γίνεται με την χρήση επιπέδων δειγματοληψίας. Σε αυτά, δημιουργείται μια έκδοση χαμηλότερης ανάλυσης ενός σήματος εισόδου που περιέχει τα σημαντικά δομικά στοιχεία χωρίς τις μικρές λεπτομέρειες που δεν είναι τόσο χρήσιμες για το δίκτυο. Επιπλέον, η χρήση δειγματοληπτικών επιπέδων συμβάλει στη μείωση του μεγέθους του δικτύου μειώνοντας τον αριθμό των παραμέτρων που προωθούνται στο επόμενο επίπεδο.

Γενικά, στα συνελικτικά νευρωνικά δίκτυα, κάθε συνελικτικό επίπεδο ακολουθείται από ένα επίπεδο δειγματοληψίας. Υπάρχουν δύο κύριες συναρτήσεις ομαδοποίησης, η μέγιστη ομαδοποίηση (max pooling) και η μέση ομαδοποίηση (average pooling).

Στη μέγιστη ομαδοποίηση υπάρχουν μέγιστα φίλτρα ομαδοποίησης τα οποία, όπως ακριβώς και τα συνελικτικά φίλτρα, είναι πίνακες που ολισθαίνουν πάνω στην εικόνα. Η διαφορά είναι ότι αυτοί οι πίνακες δεν έχουν βάρη. Το μόνο που κάνουν είναι να επιλέγουν τη μέγιστη τιμή κάθε φορά και να την περνάνε στο επόμενο επίπεδο αγνοώντας τις υπόλοιπες.

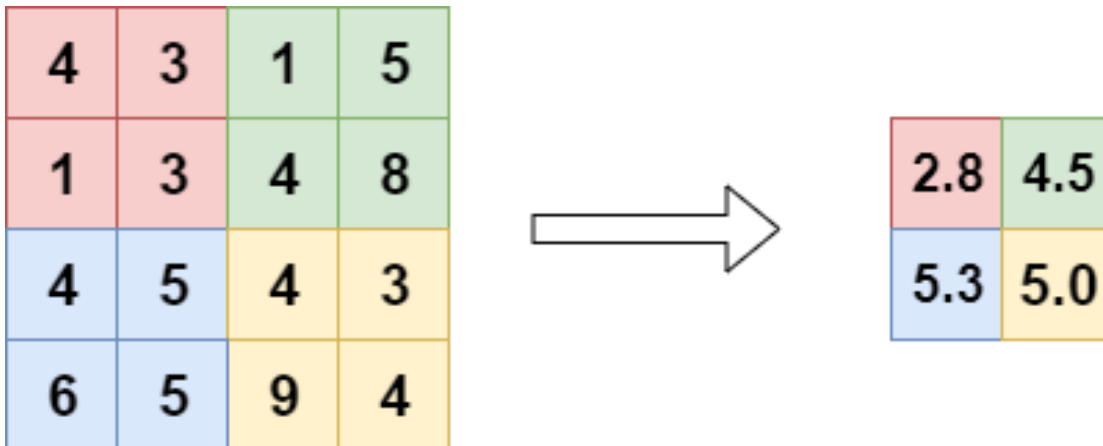
Στο Σχήμα 3.5 φαίνεται η εφαρμογή ενός φίλτρου μέγιστης ομαδοποίησης με μέγεθος  $2 \times 2$  με



Σχήμα 3.5 Φίλτρο Μέγιστης Ομαδοποίησης [25]

βήμα 2 εικονοστοιχεία σε έναν τετραγωνικό πίνακα με διαστάσεις  $4 \times 4$ . Παρατηρούμε ότι το μέγεθος της εξόδου είναι μικρότερο από την είσοδο. Αξίζει να σημειωθεί ότι όταν εφαρμόζουμε μέγιστη ομαδοποίηση στην εικόνα, το βάθος του στρώματος διατηρείται το ίδιο καθώς εφαρμόζουμε το φίλτρο ομαδοποίησης σε κάθε έναν χάρτη ξεχωριστά.

Τα φίλτρα μέσης ομαδοποίησης λειτουργούν ακριβώς με τον ίδιο τρόπο, με τη μόνη διαφορά ότι αντί να βρίσκουν το μέγιστο κάθε φορά, υπολογίζουν το μέσο όρο των τιμών στον κάθε πίνακα και τον προωθούν στην έξοδο [15]. Παρακάτω φαίνεται η ίδια εφαρμογή με πριν αλλά αυτήν τη φορά με φίλτρο μέσης ομαδοποίησης.



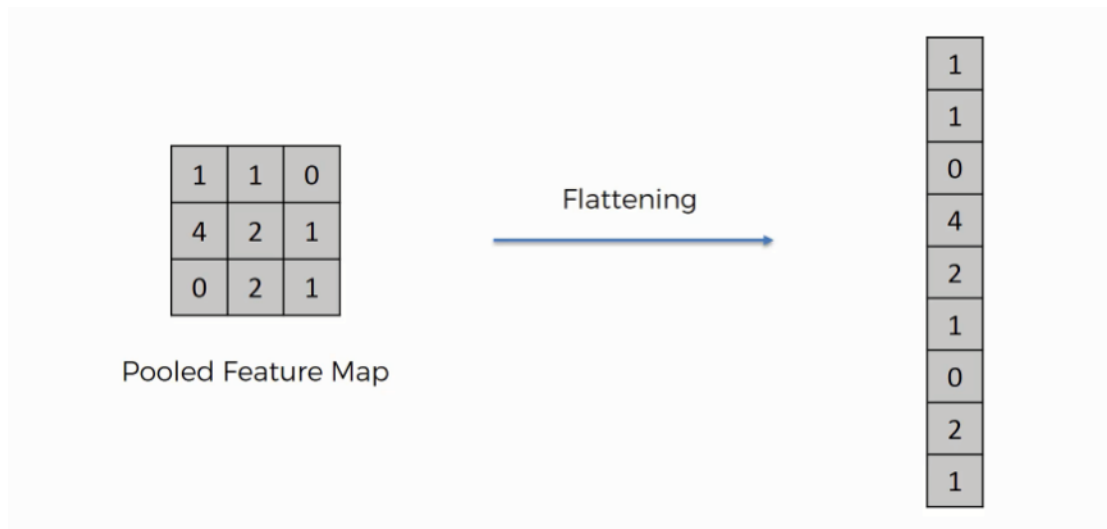
Σχήμα 3.6 Φίλτρο Μέσης Ομαδοποίησης

### 3.2.3 Πλήρως Συνδεδεμένα Επίπεδα (Fully Connected Layers)

Αφού ολοκληρωθεί η διαδικασία εξαγωγής των χαρακτηριστικών της εικόνας χρησιμοποιώντας τα συνελκτικά και δειγματοληπτικά επίπεδα, πρέπει να προχωρήσουμε στην ταξινόμηση των

εικόνων. Αυτή θα γίνει με τη χρήση ενός τεχνητού νευρωνικού δικτύου με πλήρως συνδεδεμένα επίπεδα που αναλύσαμε στο προηγούμενο κεφάλαιο [16]. Άρα ουσιαστικά η μόνη διαφορά των συνελκτικών νευρωνικών δικτύων με τα απλά έγκειται στο ότι τα συνελκτικά νευρωνικά δίκτυα η εξαγωγή των χαρακτηριστικών γίνεται αυτόματα από το δίκτυο με τη χρήση των συνελκτικών φίλτρων [17].

Ένα πρόβλημα σε αυτό είναι ότι τα νευρωνικά δίκτυα δουλεύουν με είσοδο δεδομένα με μορφή πινάκων μίας διάστασης ενώ τα συνελκτικά φίλτρα παράγουν έξοδο της οποίας οι διαστάσεις μπορεί να ποικίλουν. Για να το αντιμετωπίσουμε αυτό, προτού τροφοδοτήσουμε τους χάρτες χαρακτηριστικών στο πλήρως συνδεδεμένο δίκτυο θα πρέπει να «στρώσουμε» (flatten) την έξοδο των προηγούμενων στρωμάτων. Να μετατρέψουμε δηλαδή την πολυδιάστατη έξοδο των συνελκτικών φίλτρων σε έναν μονοδιάστατο πίνακα [15].



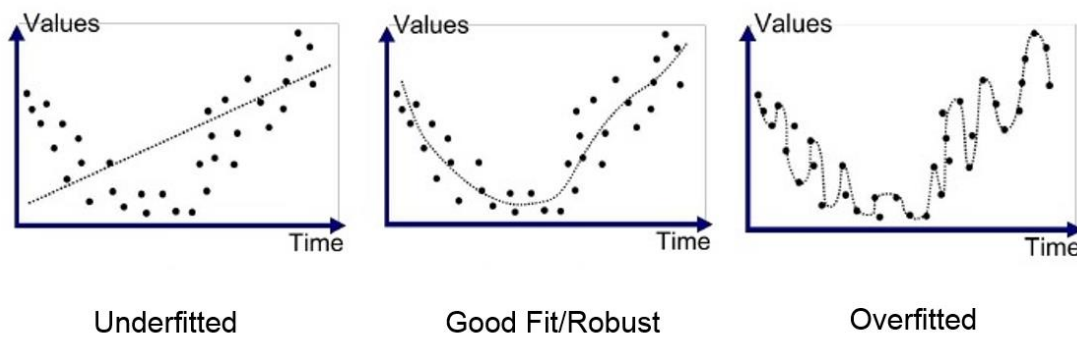
Σχήμα 3.7 Flattening

### 3.3 ΕΠΙΠΕΔΑ ΕΓΚΑΤΑΛΕΙΨΗΣ (DROPOUT LAYERS)

Οι κύριοι λόγοι που ένα μοντέλο μηχανικής μάθησης δεν επιτυγχάνει καλές προβλέψεις είναι είτε η υπερπροσαρμογή στα δεδομένα εκπαίδευσης (overfitting) είτε οι ανεπαρκείς προσαρμογές (underfitting) [19]. Στην ανεπαρκή προσαρμογή το μοντέλο δεν καταφέρνει να προσαρμοστεί στα δεδομένα εκπαίδευσης και αυτό συμβαίνει κυρίως όταν δεν έχουμε ικανοποιητικό αριθμό δεδομένων εκπαίδευσης ή όταν το μοντέλο που χρησιμοποιούμε είναι πολύ απλό για να «αποκρυπτογραφήσει» όλα τα χαρακτηριστικά που απαιτούνται για την

ταξινόμηση των δεδομένων εισόδου. Αντιθέτως, στην υπερπροσαρμογή το μοντέλο προσαρμόζεται υπερβολικά στα δεδομένα εισόδου με αποτέλεσμα να μην μαθαίνει ουσιαστικά τα χαρακτηριστικά των δεδομένων εισόδου και να μην είναι ικανό να παράγει προβλέψεις για νέα δεδομένα.

Κατά την κατασκευή μοντέλων συνελκτικών νευρωνικών δικτύων, τα μοντέλα μας είναι συνήθως βαθιά και αρκετά περίπλοκα και γι' αυτό το κύριο πρόβλημα κατά την εκπαίδευσή τους είναι η αποφυγή της υπερπροσαρμογής. Τα επίπεδα εγκατάλειψης είναι τα πιο συχνά



Σχήμα 3.8 ο Και Υποπροσαρμογή (Overfitting And Underfitting) [27]

χρησιμοποιούμενα επίπεδα για την αντιμετώπιση της υπερπροσαρμογής. Η εγκατάλειψη (dropout) είναι μια μέθοδος κανονικοποίησης για την εκπαίδευση μεγάλου αριθμού νευρωνικών δικτύων με διαφορετικές αρχιτεκτονικές παράλληλα.

Κατά τη διάρκεια της εκπαίδευσης οι νευρώνες αναπτύσσουν αλληλεξάρτηση μεταξύ τους η οποία ελέγχει την δύναμη (βάρη) του κάθε νευρώνα. Όπως έχουμε δει, όταν στοιβάζουμε δύο κρυφά συνελκτικά επίπεδα στη σειρά για την εξαγωγή χαρακτηριστικών μιας εικόνας ο πρώτος αναγνωρίζει πιο απλά σχήματα και ο επόμενος πιο σύνθετα. Εάν για παράδειγμα θέλουμε ο δεύτερος να αναγνωρίσει ένα πρόσωπο, θα πρέπει από τον πρώτο να καταλήξουν δύο κύκλοι για τα μάτια δύο ευθείες για τη μύτη και μία για το στόμα. Έτσι κάθε φορά που η είσοδος είναι αυτά τα χαρακτηριστικά η έξοδος θα είναι πρόσωπο. Με αυτόν τον τρόπο το μοντέλο οδηγείται σε υπερπροσαρμογή.

Εάν τώρα κατά τη διάρκεια της εκπαίδευσης, προσθέσουμε επίπεδα εγκατάλειψης κάποιος αριθμός νευρώνων από τα επίπεδα αγνοείται [19]. Ως αποτέλεσμα, το κάθε επίπεδο αντιμετωπίζεται σαν ένα επίπεδο με διαφορετικό αριθμό κόμβων από το προηγούμενο στρώμα. Αυτό θα έχει ως αποτέλεσμα να αναγκάσουμε το μοντέλο μας να μάθει μοτίβα τα οποία δεν



βασίζονται σε λίγα μόνο χαρακτηριστικά από τα προηγούμενα επίπεδα. Το ποσοστό των νευρώνων που αγνοείται καθορίζεται χειροκίνητα ως υπερ-παράμετρος του δικτύου κατά την κατασκευή του. Η εγκατάλειψη (dropout) είναι ένας τρόπος να είμαστε σίγουροι πως αν κάποιος κόμβος κάνει λάθος, αυτό δεν θα κυριαρχήσει στη συμπεριφορά του μοντέλου μας μιας και κανένας κόμβος δεν θα είναι πολύ ισχυρός ή αδύναμος.

### 3.4 ΒΙΒΛΙΟΘΗΚΕΣ ΚΑΙ ΠΛΑΤΦΟΡΜΕΣ ΛΟΓΙΣΜΙΚΟΥ

#### 3.4.1 Tensorflow

Η Tensorflow είναι μια ελεύθερη βιβλιοθήκη ανοιχτού κώδικα για τη δημιουργία εφαρμογών μηχανικής μάθησης. Πρόκειται για μια συμβολική βιβλιοθήκη μαθηματικών που χρησιμοποιεί ροή δεδομένων για την εκτέλεση εργασιών που βοηθούν στη δημιουργία και την εκπαίδευση τεχνητών νευρωνικών δικτύων.

Η Tensorflow δημιουργήθηκε από ερευνητές της Google και είναι η πιο διάσημη βιβλιοθήκη μηχανικής μάθησης σήμερα καθώς δίνει τη δυνατότητα σε όποιον θέλει να ασχοληθεί να δημιουργήσει γραφήματα ροής δεδομένων και να καθορίσει τον τρόπο με τον οποίο τα δεδομένα κινούνται μέσω αυτών λαμβάνοντας ως εισόδους μια πολυδιάστατη συστοιχία που ονομάζεται tensor. Ένας tensor είναι ένας φορέας διαστάσεων που μπορεί να αντιπροσωπεύει όλους τους τύπους δεδομένων είτε πρόκειται για εικόνες είτε για απλούς αριθμούς και αντιπροσωπεύει τόσο τα δεδομένα εισόδου όσο και τις εξόδους των νευρώνων μέσα στο δίκτυο. Η ονομασία Tensorflow προέρχεται από το tensor και το flow που σημαίνει ροή. Άρα ουσιαστικά η Tensorflow αντιπροσωπεύει τη ροή των tensor μέσα από τα γραφήματα που δημιουργούμε.

Ένας ακόμα λόγος που η Tensorflow είναι η πιο χρησιμοποιούμενη βιβλιοθήκη εφαρμογών μηχανικής μάθησης είναι ότι υποστηρίζει πολλαπλές επιλογές εκτέλεσης τόσο στην CPU όσο και στην GPU. Αυτό δίνει το πλεονέκτημα στα δίκτυά μας να εκπαιδεύονται σε μικρότερο χρονικό διάστημα. Επίσης, επιτρέπει στον προγραμματιστή να απεικονίσει την κατασκευή του νευρωνικού του δικτύου κάτι που διευκολύνει πολύ την ανίχνευση και τη διόρθωση σφαλμάτων του προγράμματος.

#### 3.4.2 Keras

Η Keras είναι μια βιβλιοθήκη ανοιχτού λογισμικού χτισμένη πάνω στην Tensorflow. Παρέχει μια διεπαφή προγραμματισμού εφαρμογών (API), η οποία καθορίζει τις αλληλεπιδράσεις μεταξύ εφαρμογών λογισμικού (software) και εξαρτημάτων του ηλεκτρονικού υπολογιστή (hardware),

για την κατασκευή νευρωνικών δικτύων. Η Keras επιτρέπει τη γρήγορη δημιουργία νευρωνικών δικτύων χωρίς τον υπολογισμό των μαθηματικών που κρύβονται πίσω από τους tensor.

Η Keras δίνει τη δυνατότητα ακόμα και σε αρχάριους προγραμματιστές να ασχοληθούν με τα νευρωνικά δίκτυα καθώς υποστηρίζει τη δημιουργία όλων των μοντέλων νευρωνικών δικτύων. Επίσης, είναι ιδιαίτερα εύχρηστη και ευανάγνωστη αφού κάθε γραμμή κώδικα δημιουργεί ένα επίπεδο δικτύου. Τέλος, μπορεί να τρέχει και αυτή τόσο στην CPU όσο και στην GPU.

#### 3.4.3 NumPy

Η NumPy είναι μια βιβλιοθήκη η οποία είναι γραμμένη σε Python αλλά τα μέρη που απαιτούν γρήγορους υπολογισμούς είναι γραμμένα σε C ή C++. Όπως έχουμε δει, τα νευρωνικά δίκτυα δουλεύουν κυρίως με τη χρήση πινάκων. Η NumPy είναι η βιβλιοθήκη που χρησιμοποιείται για την εκτέλεση εργασιών γραμμικής άλγεβρας, μετασχηματισμών fourier και πινάκων.

Η Python για αυτές τις εργασίες διαθέτει τις λίστες. Ωστόσο, οι συστοιχίες της NumPy είναι πολύ πιο αποτελεσματικές και γρήγορες καθώς αποθηκεύονται σε μια συνεχή θέση μνήμης. Επίσης, η NumPy είναι φτιαγμένη έτσι ώστε να βελτιστοποιείται σε λειτουργίες με τις πιο πρόσφατες αρχιτεκτονικές CPU.

#### 3.4.4 OpenCV

Η OpenCV είναι μια βιβλιοθήκη ανοιχτού κώδικα με πολλά χρήσιμα εργαλεία για την επεξεργασία εικόνων. Διαθέτει περισσότερους από 2500 αλγορίθμους που διευκολύνουν στην μηχανική όραση. Με την χρήση της OpenCV, γίνονται εύκολες κάποιες αρκετά περίπλοκες εργασίες όπως η αναγνώριση προσώπων και αντικειμένων, η εισαγωγή βίντεο στα μοντέλα μας, ο συνδυασμός εικόνων και πολλές άλλες. Πλέον, στις περισσότερες εφαρμογές που απαιτούν επεξεργασία εικόνας γίνεται χρήση της OpenCV καθώς είναι πάρα πολύ εύκολη στη χρήση.

### 3.5 ΥΛΟΠΟΙΗΣΗ ΣΥΝΕΛΙΚΤΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΜΕ GPU

Τα συνελικτικά νευρωνικά δίκτυα πλέον χρησιμοποιούνται για την επίλυση απλών αλλά και πολύπλοκων διεργασιών. Τα μεγαλύτερα και βαθύτερα συνελικτικά νευρωνικά δίκτυα, με περισσότερες παραμέτρους, συνήθως πετυχαίνουν καλύτερη ακρίβεια. Όμως, όσες περισσότερες παραμέτρους έχει ένα δίκτυο τόσο περισσότερη υπολογιστική ισχύς απαιτείται. Γι' αυτόν το λόγο, η εκπαίδευση των συνελικτικών δικτύων που συνήθως έχουν πολλές παραμέτρους γίνεται με χρήση της κάρτας γραφικών (GPU).

Η GPU είναι ένας εξειδικευμένος επεξεργαστής που εκτελεί συμβατικά τις λειτουργίες κινητής υποδιαστολής που απαιτούνται για την απόδοση γραφικών. Χρησιμοποιείται δηλαδή για εκτεταμένους υπολογισμούς γραφικών και μαθηματικών. Η κύρια διαφορά μεταξύ GPU και CPU είναι ότι οι GPU αφιερώνουν περισσότερα τρανζίστορ σε αριθμητικές λογικές μονάδες και λιγότερα σε κρυφές μνήμες και έλεγχο ροής. Η εφαρμογή συνελκτικών φίλτρων, η ανανέωση των βαρών και γενικότερα όλες οι λειτουργίες των συνελκτικών νευρωνικών δικτύων απαιτούν πολλούς μαθηματικούς υπολογισμούς. Έτσι, είναι απαραίτητη η χρήση της GPU για την εκπαίδευση αυτών των δικτύων.

Το Google Colab [21] είναι μία δωρεάν υπηρεσία της Google για προγραμματιστές τεχνητής νοημοσύνης. Με το Colab μπορεί ο καθένας να αναπτύξει εφαρμογές βαθιάς μάθησης σε γλώσσα python χρησιμοποιώντας δωρεάν GPU. Η κάρτα γραφικών που παρέχει η Colab είναι η NVIDIA Tesla K80 και μπορεί να χρησιμοποιηθεί για έως και 12 ώρες συνεχόμενα. Οι δυνατότητες που μας παρέχει η Colab είναι υπεραρκετές για την ανάπτυξη όλων των εφαρμογών αυτής της εργασίας και έτσι, η εκπαίδευση όλων των δικτύων έγινε στο περιβάλλον της Google Colab.

## ΚΕΦΑΛΑΙΟ 4ο

---

### Εντοπισμός Αντικειμένων Και Υπολογισμός Αποστάσεων

#### 4.1 ΕΙΣΑΓΩΗ

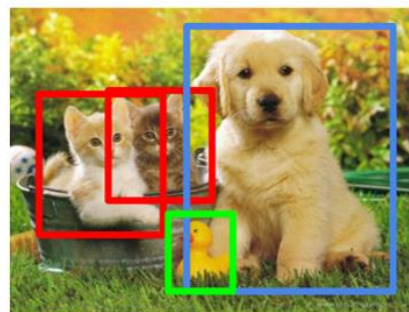
Ο εντοπισμός διαφόρων αντικειμένων στον δρόμο κατά τη διάρκεια της οδήγησης και ο υπολογισμός της απόστασης στην οποία βρίσκονται αποτελεί ίσως την πιο σημαντική εργασία στην αυτόνομη οδήγηση. Μέχρι τώρα, στα προηγούμενα κεφάλαια, είδαμε πως τα βαθιά νευρωνικά δίκτυα μπορούν να χρησιμοποιηθούν για την ταξινόμηση των εικόνων. Η διαφορά της ταξινόμησης και του εντοπισμού των αντικειμένων βρίσκεται στο ότι στην ταξινόμηση υπάρχει μόνο ένα αντικείμενο στην κάθε εικόνα και προσπαθεί το μοντέλο μας να το αναγνωρίσει ενώ στον εντοπισμό των αντικειμένων, το μοντέλο προσπαθεί όχι μόνο να ταξινομήσει το κάθε αντικείμενο σε κλάσεις αλλά να βρει και τις ακριβείς συντεταγμένες του στην εικόνα [16].

#### Classification



CAT

#### Object Detection



CAT, DOG, DUCK

Σχήμα 4.1 Διαφορά Μεταξύ Ταξινόμησης Εικόνων Και Εύρεσης Αντικειμένων

Εύκολα μπορεί να αντιληφθεί κανείς τη σπουδαιότητα του εντοπισμού αντικειμένων στην αυτόνομη οδήγηση. Για παράδειγμα, είναι πολύ σημαντικό να είναι σε θέση το σύστημά μας να εντοπίζει με ακρίβεια άλλα αυτοκίνητα, πεζούς ή ζώα που μπορεί να βρεθούν στο δρόμο. Ωστόσο, αυτό δεν είναι αρκετό. Πέρα από τον εντοπισμό των αντικειμένων, το σύστημα θα πρέπει να είναι σε θέση να προσδιορίσει και την απόσταση αυτών από το όχημά μας έτσι ώστε να προβεί στις απαραίτητες ενέργειες όπως το φρενάρισμα.

#### 4.2 ΑΛΓΟΡΙΘΜΟΣ YOLO (YOU ONLY LOOK ONCE)

Ο αλγόριθμος YOLO έχει σχεδιαστεί έτσι ώστε να μπορεί να επιτύχει τον εντοπισμό των αντικειμένων σε πραγματικό χρόνο [17]. Σε αντίθεση με άλλους αλγορίθμους εντοπισμού αντικειμένων, ο YOLO είναι πολύ πιο γρήγορος επιτυγχάνοντας πολύ καλά ποσοστά προβλέψεων. Αυτός είναι και ο λόγος για τον οποίο είναι πολύ διαδεδομένος σε διαδικασίες εντοπισμού αντικειμένων.

Οι δημιουργοί του YOLO υιοθέτησαν μια διαφορετική προσέγγιση από τα υπόλοιπα μοντέλα εντοπισμού. Σε αντίθεση με άλλα μοντέλα που εκτελούν ανίχνευση σε διαφορετικές περιοχές και καταλήγουν να εκτελούν προβλέψεις πολλές φορές για διάφορες περιοχές της εικόνας, η αρχιτεκτονική YOLO μοιάζει με ένα πλήρως συνδεδεμένο συνελκτικό νευρωνικό δίκτυο που περνά την εικόνα μόνο μία φορά (εξ ου και το όνομα YOU ONLY LOOK ONCE) και η έξοδος είναι ένας πίνακας πρόβλεψης.

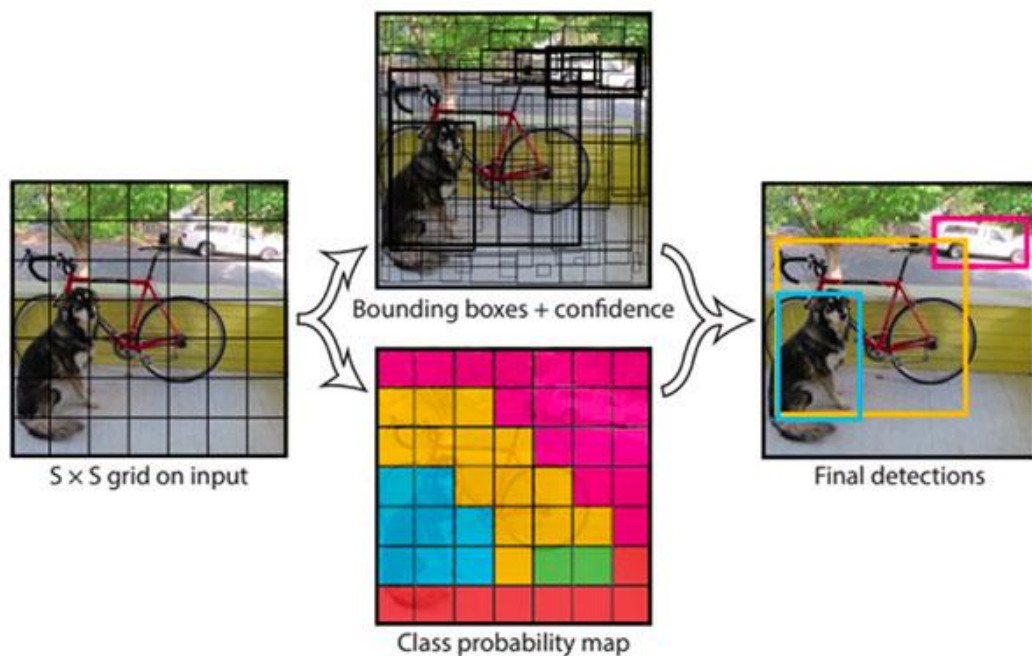
Ο αλγόριθμος YOLO χωρίζει την εικόνα εισόδου σε ένα πλέγμα  $S \times S$ . Εάν το κέντρο ενός αντικειμένου πέσει σε ένα κελί πλέγματος, τότε αυτό το κελί πλέγματος είναι υπεύθυνο για την ανίχνευση αυτού του αντικειμένου. Κάθε κελί πλέγματος προβλέπει τα πλαίσια του και βαθμολογίες εμπιστοσύνης γι' αυτά τα πλαίσια. Αυτές οι βαθμολογίες δείχνουν πόσο σίγουρο είναι το μοντέλο ότι το πλαίσιο αυτό περιέχει ένα αντικείμενο και πόσο ακριβές είναι αυτό το πλαίσιο. Εάν δεν υπάρχει αντικείμενο σε κάποιο κελί, η βαθμολογία εμπιστοσύνης πρέπει να είναι ίση με μηδέν. Διαφορετικά, η βαθμολογία εμπιστοσύνης θα πρέπει να ισούται με τη βαθμολογία IOU.

Η βαθμολογία IOU (INTERSECTION OVER UNION) αξιολογεί την επικάλυψη μεταξύ του προβλεπόμενου πλαισίου και του αναμενόμενου πλαισίου. Στο Σχήμα 4.3 φαίνεται η διαφορά μιας κακής με μέτρια και μία καλή πρόβλεψη σύμφωνα με τη μέτρηση βαθμολογίας IOU



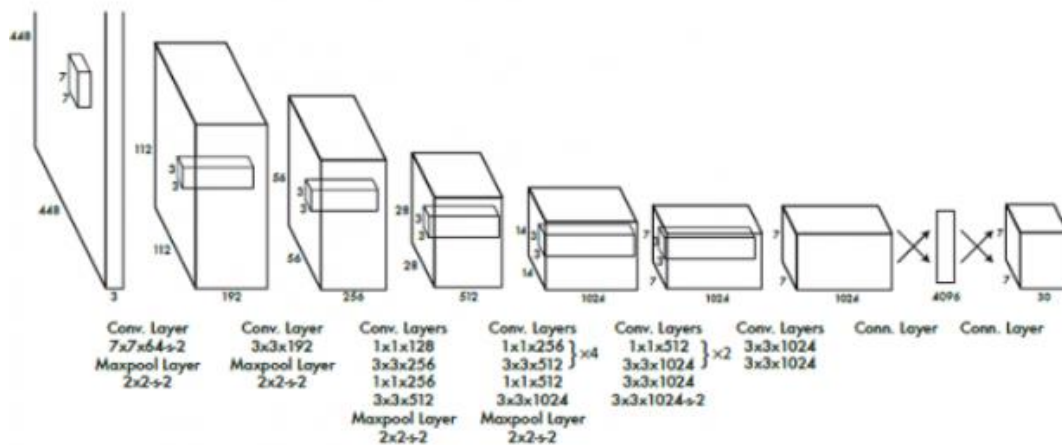
Σχήμα 4.3 Αξιολόγηση Πρόβλεψης Με Βαθμολογία IOU [30]

Κάθε πλαίσιο οριοθέτησης στον αλγόριθμο YOLO αποτελείται από πέντε προβλέψεις, τις  $x$ ,  $y$ ,  $w$ ,  $h$  και την εμπιστοσύνη. Οι συντεταγμένες  $(x, y)$  αντιπροσωπεύουν το κέντρο του πλαισίου, τα  $w$ ,  $h$  είναι το πλάτος και το ύψος του πλαισίου αντίστοιχα και εκφράζονται ως ποσοστά της συνολικής εικόνας και τέλος η πρόβλεψη εμπιστοσύνης αντιπροσωπεύει τη μέτρηση IOU μεταξύ του προβλεπόμενου πλαισίου και οποιουδήποτε αναμενόμενου πλαισίου εξόδου. Επίσης, κάθε κελί πλέγματος προβλέπει πιθανότητες για την κλάση στην οποία ανήκει το αντικείμενο που περικλείει. Στο τέλος, πολλαπλασιάζονται οι πιθανότητες πρόβλεψης των αντικειμένων με την μέτρηση IOU και κρατάμε το πλαίσιο με την πιο ισχυρή πρόβλεψη



Σχήμα 4.2 Εξαγωγή Προβλέψεων Με Χρήση Του Αλγορίθμου YOLO

Το δίκτυο του YOLO έχει 24 συνελκτικά επίπεδα ακολουθούμενα από 2 πλήρως συνδεδεμένα επίπεδα για την ταξινόμηση. Ως επίπεδα μείωσης χρησιμοποιούνται  $1 \times 1$  ακολουθούμενα από  $3 \times 3$  συνελκτικά επίπεδα. Η τελική έξοδος είναι ένας  $7 \times 7 \times 30$  tensor.



Σχήμα 4.4 Δομή Του Δικτύου YOLO

### 4.3 ΕΝΤΟΠΙΣΜΟΣ ΑΝΤΙΚΕΙΜΕΝΩΝ

Όσον αφορά τον εντοπισμό και την αναγνώριση των αντικειμένων σε μία εικόνα, θα μπορούσαμε να εκπαιδεύσουμε ένα μοντέλο YOLO σε ένα σετ δεδομένων που θα περιέχει εικόνες δρόμου με αντικείμενα που θέλουμε να προβλέπουμε μέσα σε πλαίσια για τα οποία γνωρίζουμε τις συντεταγμένες και το αντικείμενο που περιέχουν. Ωστόσο, για καλύτερα αποτελέσματα, σε αυτό το κεφάλαιο θα χρησιμοποιήσουμε ένα προ-εκπαιδευμένο μοντέλο YOLO v5, που αποτελεί το πιο πρόσφατο μοντέλο YOLO, της Ultralytics [18] το οποίο είναι εκπαιδευμένο πάνω στο σετ δεδομένων COCO που αποτελείται από ένα σύνολο πάνω από 250.000 εικόνων ώστε να εντοπίζει 80 διαφορετικά αντικείμενα που συναντάμε καθημερινά. Τα αντικείμενα τα οποία μπορεί να εντοπίζει και να ταξινομεί φαίνονται στο Σχήμα 4.5.

1 person	21 elephant	41 wine glass	61 diningtable
2 bicycle	22 bear	42 cup	62 toilet
3 car	23 zebra	43 fork	63 tvmonitor
4 motorbike	24 giraffe	44 knife	64 laptop
5 aeroplane	25 backpack	45 spoon	65 mouse
6 bus	26 umbrella	46 bowl	66 remote
7 train	27 handbag	47 banana	67 keyboard
8 truck	28 tie	48 apple	68 cell phone
9 boat	29 suitcase	49 sandwich	69 microwave
10 traffic light	30 frisbee	50 orange	70 oven
11 fire hydrant	31 skis	51 broccoli	71 toaster
12 stop sign	32 snowboard	52 carrot	72 sink
13 parking meter	33 sports ball	53 hot dog	73 refrigerator
14 bench	34 kite	54 pizza	74 book
15 bird	35 baseball bat	55 donut	75 clock
16 cat	36 baseball glove	56 cake	76 vase
17 dog	37 skateboard	57 chair	77 scissors
18 horse	38 surfboard	58 sofa	78 teddy bear
19 sheep	39 tennis racket	59 pottedplant	79 hair drier
20 cow	40 bottle	50 bed	80 toothbrush

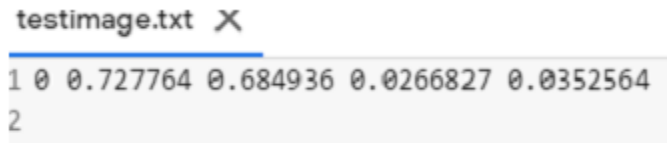
Σχήμα 4.5 Οι 80 Κλάσεις Του Σετ Δεδομένων COCO

Όπως φαίνεται στην παραπάνω εικόνα, τα περισσότερα αντικείμενα που εντοπίζει δεν τα συναντάμε την ώρα της οδήγησης. Παρόλα αυτά, υπάρχουν και τα βασικά αντικείμενα που θέλουμε να αναγνωρίζει ένα αυτόνομο όχημα όπως άνθρωποι, ποδήλατα, αυτοκίνητα, σκυλιά και τα λοιπά. Στο Σχήμα 4.6 φαίνεται η εικόνα εξόδου του προ-εκπαιδευμένου YOLO δικτύου για μια εικόνα εισόδου.

Παρατηρούμε ότι στα αντικείμενα που έχει εντοπίσει έχει προσθέσει ένα πλαίσιο γύρω τους και πάνω από το καθένα γράφει την κλάση στην οποία πιστεύει ότι ανήκει το κάθε αντικείμενο και πόσο σίγουρο είναι ότι το έχει ταξινομήσει σωστά. Εκτός από την εικόνα πρόβλεψης, ο αλγόριθμος YOLO επιστρέφει και ένα αρχείο μορφής .txt το οποίο περιέχει όλες τις προβλέψεις του.

Σχήμα 4.6 Εφαρμογή Προεκπαιδευμένου Μοντέλου YOLO





```
testimage.txt X
1 0 0.727764 0.684936 0.0266827 0.0352564
2
```

Σχήμα 4.7 Αρχείο .txt Εξόδου Του YOLO

Στο Σχήμα 4.7 φαίνεται ένα παραγόμενο αρχείο .txt στο οποίο το μοντέλο μας έχει προβλέψει ένα αντικείμενο της κλάσης 0, το πλαίσιο του οποίου έχει κέντρο στο 72% του συνολικού πλάτους της εικόνας και 68% του συνολικού ύψους της εικόνας. Το πλάτος του πλαισίου είναι ίσο με το 0,26% του συνολικού πλάτους και το ύψος ίσο με το 0,35% του συνολικού ύψους.

#### 4.4 ΥΠΟΛΟΓΙΣΜΟΣ ΑΠΟΣΤΑΣΕΩΝ

Για τον υπολογισμό της απόστασης που βρίσκεται το κάθε αντικείμενο από την κάμερα θα χρησιμοποιηθεί μια παραλλαγή του τύπου του εστιακού μήκους:

$$\text{focalLength} = (\text{monitorWidth} * \text{Distance}) / \text{Width} \quad (4.1)$$

όπου

focalLength : Εστιακή απόσταση

monitorWidth : Το πλάτος ενός αντικειμένου της εικόνας σε pixel

Distance : Η απόσταση του αντικειμένου από τον φακό

Αρχικά, θα πρέπει να υπολογίσουμε την εστιακή απόσταση της συσκευής μας. Για να γίνει αυτό, χρησιμοποιούμε μια εικόνα ενός αντικειμένου γνωστού πλάτους που μπορεί να εντοπίσει το προ-εκπαιδευμένο δίκτυό μας, τραβηγμένη από μια γνωστή απόσταση. Width : Το πλάτος του αντικειμένου στην πραγματικότητα.



Σχήμα 4.8 Εικόνα Απεικόνισης Αντικειμένων Με Γνωστή Απόσταση Από Τον Φακό

Αυτή η εικόνα είναι τραβηγμένη 1 μέτρο μακριά από την οθόνη του υπολογιστή η οποία έχει μήκος 35 εκατοστά. Οπότε, για τον υπολογισμό της εστιακής απόστασης, μένει να βρούμε το πλάτος της οθόνης σε pixel. Προκειμένου να το υπολογίσουμε αυτό, θα προωθήσουμε την εικόνα στο προ-εκπαιδευμένο μοντέλο του Yolo και θα το υπολογίσουμε με την βοήθεια του παραγόμενου .txt αρχείου υπό την προϋπόθεση βέβαια ότι θα εντοπίσει την οθόνη.

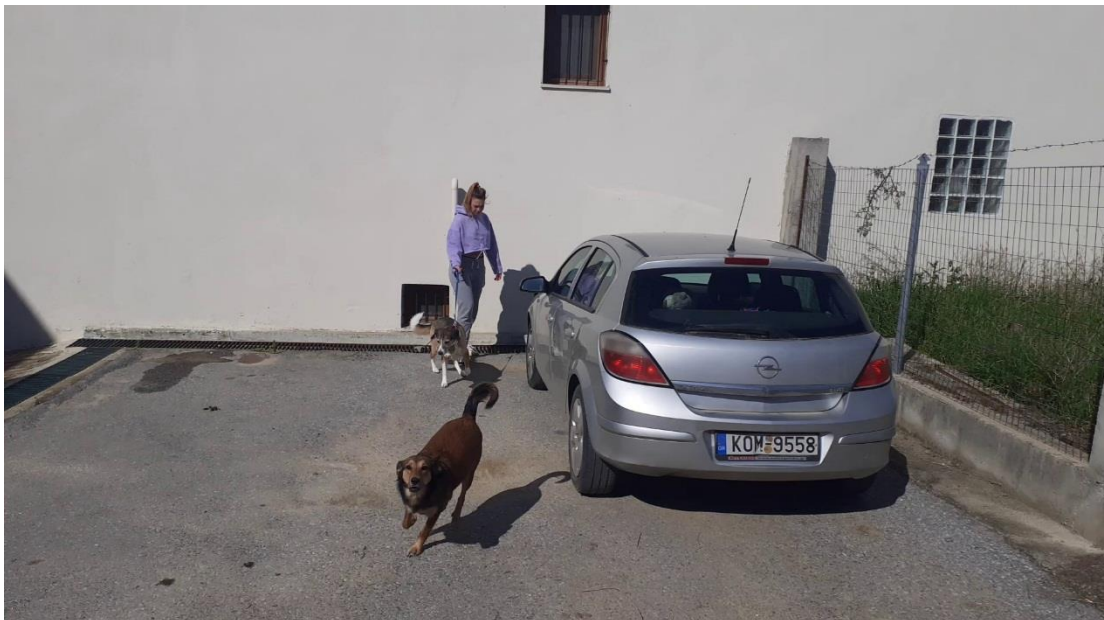


Σχήμα 4.9 Εντοπισμός Αντικειμένων Στο Σχήμα 4.8

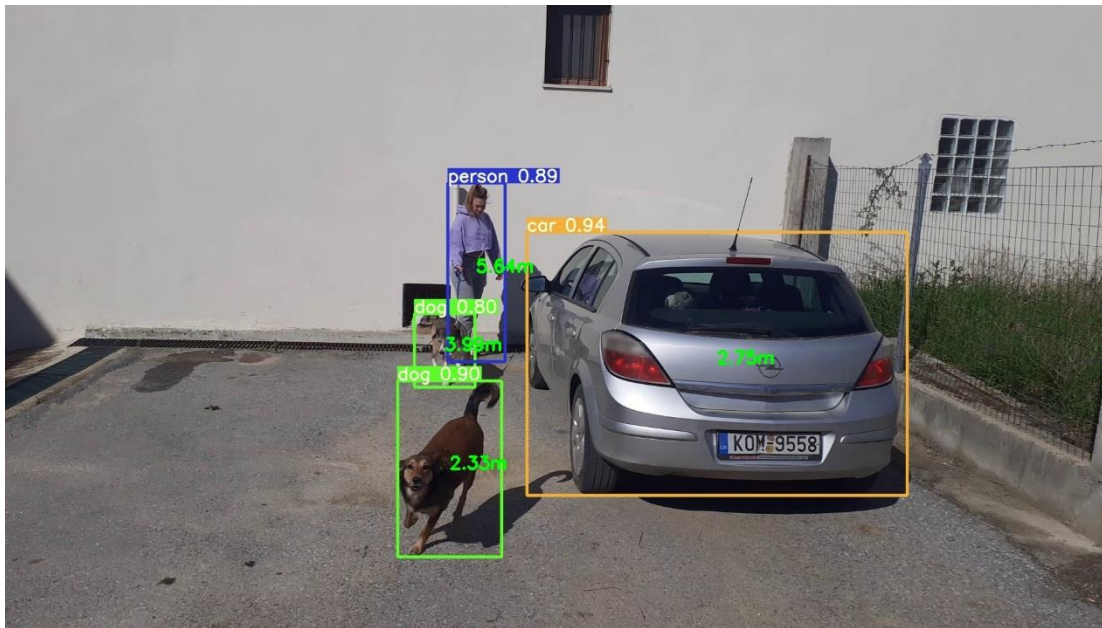
Όπως φαίνεται στην εικόνα, το Yolo εντόπισε την οθόνη του υπολογιστή καθώς επίσης ακόμα 3 αντικείμενα. Γνωρίζοντας το συνολικό πλάτος της εικόνας, το πολλαπλασιάζουμε με την τέταρτη τιμή του αρχείου .txt που αφορά την οθόνη και βρίσκουμε το πλάτος της οθόνης σε pixels. Έπειτα, υπολογίζουμε την τιμή της εστιακής απόστασης από την εξίσωσή της.

Τώρα που γνωρίζουμε την τιμή της εστιακής απόστασης, ο υπολογισμός των αποστάσεων για νέα αντικείμενα είναι πολύ απλός. Το μόνο που έχουμε να κάνουμε είναι να ορίσουμε το πραγματικό πλάτος των αντικειμένων που θέλουμε να υπολογίσουμε την απόσταση από την κάμερα, να περάσουμε την εικόνα μας από το προ-εκπαιδευμένο μοντέλο έτσι ώστε να πάρουμε το αρχείο των προβλέψεων και να χρησιμοποιήσουμε τον τύπο της εστιακής απόστασης για το καθένα από αυτά.

Στον κώδικά μας, ορίζουμε τα μέσα πλάτη των αντικειμένων που θέλουμε να υπολογίζουμε την απόστασή τους από την κάμερα, μερικά από αυτά είναι άνθρωπος, αυτοκίνητο, μηχανή, ποδήλατο, σκύλος, γάτα, φορτηγό, λεωφορείο, φωτεινός σηματοδότης. Στις παρακάτω εικόνες φαίνονται τα αποτελέσματα όλης της διαδικασίας για μία εικόνα εισόδου.



Σχήμα 4.10 Παράδειγμα Εικόνας Εισόδου Υπολογισμού Αποστάσεων



Σχήμα 4.11 Παράδειγμα Εικόνας Εξόδου Υπολογισμού Αποστάσεων

Ένα πρόβλημα που μπορεί εύκολα να προκύψει από αυτόν τον τρόπο υπολογισμού των αποστάσεων είναι ότι χρησιμοποιούμε ένα μέσο πλάτος για το κάθε αντικείμενο και όχι το ακριβές. Επίσης, όταν το μοντέλο μας εντοπίζει ένα λεωφορείο για παράδειγμα, θέτει ένα πλαίσιο γύρω από αυτό και σαν πλάτος του λεωφορείου μας δίνει το πλάτος του πλαισίου.

Αυτό όμως μπορεί να επηρεάσει αισθητά το πλάτος του αντικειμένου που προβλέπει το μοντέλο μας. Όπως φαίνεται στο Σχήμα 4.12 το πλάτος του κάθε αντικειμένου μπορεί να είναι διαφορετικό κάθε φορά και εξαρτάται κάθε φορά από τη γωνία που βλέπουμε το αντικείμενο. Συνεπώς, χρησιμοποιώντας την εξίσωση της εστιακής απόστασης, θα παίρναμε ως αποτέλεσμα ότι το λεωφορείο βρίσκεται πολύ πιο κοντά απ' ότι είναι στην πραγματικότητα στις πρώτες δύο περιπτώσεις που φαίνονται στο Σχήμα 4.12.



Σχήμα 4.12 Λόγος Εσφαλμένης Εκτίμησης της Απόστασης

## ΚΕΦΑΛΑΙΟ 5ο

# Αναγνώριση κυκλοφοριακής σήμανσης και φωτεινών σηματοδοτών

### 5.1 ΕΙΣΑΓΩΓΗ

Κατά τη διάρκεια της οδήγησης ένας οδηγός είναι απαραίτητο να παρατηρεί, να αναγνωρίζει και να προσαρμόζει τις ενέργειές του ανάλογα με την κυκλοφοριακή σήμανση και τους φωτεινούς σηματοδότες. Αυτό, αποτελεί και μια πρόκληση για τους επιστήμονες που ασχολούνται με την δημιουργία αυτόνομων οχημάτων.

Η αναγνώριση της κυκλοφοριακής σήμανσης δεν είναι μία απλή διαδικασία. Αρχικά, είναι ένα πρόβλημα εντοπισμού και όχι απλής ταξινόμησης. Αυτό σημαίνει πως απαιτείται μεγάλος αριθμός δεδομένων και υπολογιστικών πόρων ώστε να εκπαιδευτεί αποτελεσματικά το δίκτυο μας. Πέρα από αυτό, υπάρχουν και λειτουργικά προβλήματα που μπορεί να προκύψουν. Ένα από αυτά είναι ότι η κυκλοφοριακή σήμανση δεν είναι ίδια σε όλες τις χώρες. Μπορεί στις περισσότερες ευρωπαϊκές χώρες η σήμανση να μοιάζει και ένας άνθρωπος να μπορεί να αντιληφθεί τις βασικές πινακίδες, ωστόσο, εάν αλλάξει το σχήμα για παράδειγμα, ο υπολογιστής δεν θα είναι σε θέση να τις αναγνωρίσει.

Όσον αφορά τον εντοπισμό και αναγνώριση των φωτεινών σηματοδοτών, η διαδικασία είναι πολύ πιο απλή και δεν προκύπτουν τόσα πολλά προβλήματα. Είναι όμως απαραίτητο οι προβλέψεις να γίνονται γρήγορα και με ακρίβεια.

### 5.2 ΤΑΞΙΝΟΜΗΣΗ ΚΥΚΛΟΦΟΡΙΑΚΗΣ ΣΗΜΑΝΣΗΣ

Σκοπός αυτής της ενότητας είναι η δημιουργία ενός δικτύου που θα ταξινομεί τις πινακίδες κυκλοφοριακής σήμανσης σε κλάσεις. Θα δέχεται δηλαδή ως είσοδο μια εικόνα που απεικονίζει μόνο μια πινακίδα και θα είναι σε θέση να αναγνωρίζει τι ακριβώς συμβολίζει.

### 5.2.1 Σύνολο Δεδομένων (Dataset)

Το σύνολο δεδομένων που επιλέχθηκε για την εκπαίδευση του δικτύου είναι το πιο διάσημο ελεύθερα διαθέσιμο σύνολο δεδομένων για την αναγνώριση της κυκλοφοριακής σήμανσης. Το German Traffic Sign Recognition Benchmark (GTSRB) [19] είναι ένα σύνολο δεδομένων που δημιουργήθηκε από δέκα ώρες βίντεο οδήγησης σε δρόμους της Γερμανίας. Το GTSRB περιέχει 51.839 εικόνες εκ των οποίων οι 39.209 χρησιμοποιούνται για την εκπαίδευση(training) του δικτύου ενώ οι υπόλοιπες 12.630 χρησιμοποιούνται για τον έλεγχο(test). Οι διαστάσεις των εικόνων δεν είναι ίδιες αλλά εκτείνονται από 15 x 15 έως 250 x 250 pixels.



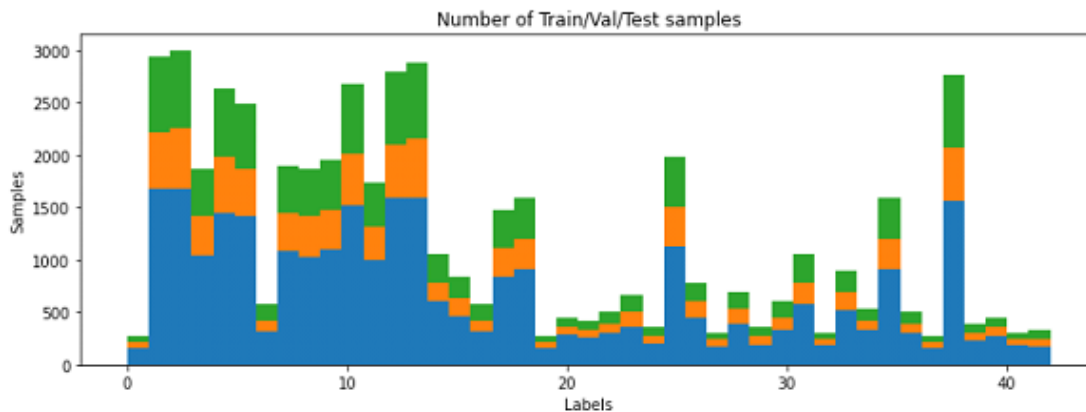
Σχήμα 5.1 Οι 43 Κλάσεις του GTSRB

Οι εικόνες χωρίζονται σε 43 διαφορετικές κλάσεις, που κάθε μια από αυτές αντιπροσωπεύει έναν ξεχωριστό τύπο πινακίδας. Όπως παρατηρούμε από το Σχήμα 5.1 οι πινακίδες δεν είναι ίδιες με αυτές του Ελληνικού δικτύου κυκλοφορίας αλλά μοιάζουν αρκετά.



### 5.2.2 Προ-Επεξεργασία Δεδομένων

Όπως είδαμε σε προηγούμενο κεφάλαιο, προκειμένου να επιλύσουμε το πρόβλημα με τη χρήση συνελκτικών νευρωνικών δικτύων, θα πρέπει να διακρίνουμε το σύνολο των δεδομένων εκπαίδευσης σε δύο μεταβλητές. Η μία θα περιέχει τις εικόνες και η άλλη τις ετικέτες για την κάθε εικόνα ξεχωριστά. Έπειτα, το σύνολο των εικόνων χωρίζεται σε 29406 εικόνες για την εκπαίδευση (training) του νευρωνικού δικτύου, 9803 εικόνες για επικύρωση (validation) και 12630 εικόνες για την δοκιμή (testing).



Σχήμα 5.2 Δεδομένα Εκπαίδευσης Ανά κλάση

Στην παραπάνω εικόνα με μπλε χρώμα απεικονίζεται ο αριθμός των εικόνων εισόδου που έχουμε για κάθε κλάση και προορίζονται για εκπαίδευση, με πορτοκαλί για επικύρωση και με πράσινο για έλεγχο.

Απαραίτητο βήμα για την τροφοδοσία των εικόνων στο νευρωνικό δίκτυο είναι ο ανασχηματισμός των εικόνων σε ίδιες διαστάσεις. Γι' αυτό με την βοήθεια του OpenCV μετατρέπουμε τις διαστάσεις όλων των εικόνων σε 32 x 32 x 3.

```
Data shapes
train(29406, 32, 32, 3)
validation(9803, 32, 32, 3)
test(12630, 32, 32, 3)
```

Στη συνέχεια, προκειμένου να βελτιωθεί η ποιότητα των εικόνων που τροφοδοτούνται στο

Σχήμα 5.3 Διαστάσεις Εικόνων

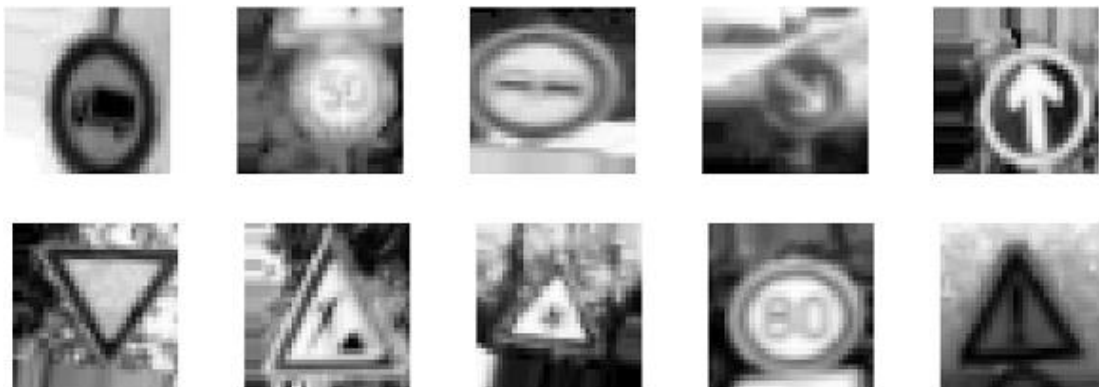
δίκτυο και να αυξηθεί η ταχύτητα εκπαίδευσης του νευρωνικού δικτύου, εξαλείφουμε τον



θόρυβο, τις μετατρέπουμε σε ασπρόμαυρες, ομαλοποιούμε τον φωτισμό και κανονικοποιούμε όλα τα δεδομένα από 0 έως 1.

Τέλος, επειδή παρατηρούμε πως σε μερικές κλάσεις ο αριθμός των δειγμάτων που έχουμε είναι μικρός χρησιμοποιούμε την τεχνική data augmentation έτσι ώστε να αυξήσουμε τα δεδομένα μας και να τα κάνουμε πιο γενικά.

Η τεχνική data augmentation χρησιμοποιείται για να αποφύγουμε την υπερπροσαρμογή στα δεδομένα εκπαίδευσης (overfitting). Αυτό γίνεται με την αύξηση των δεδομένων μιας και η τεχνική data augmentation περιλαμβάνει την περιστροφή, το ζουμάρισμα, την αλλαγή του φωτισμού και άλλους μετασχηματισμούς που εφαρμόζονται στην κάθε εικόνα. Ως αποτέλεσμα, από την κάθε εικόνα του συνόλου δεδομένων μας, παίρνουμε κι άλλες νέες εικόνες που απεικονίζουν το ίδιο απλά έχουν υποστεί κάποια επεξεργασία. Η τεχνική αυτή επιτρέπει στο δίκτυο να «βλέπει» πολλές διαφορετικές παραλλαγές του αντικειμένου μειώνοντας έτσι την εξάρτησή του από το πρωτότυπο κατά τη διάρκεια της εκπαίδευσης. Μερικά από τα αποτελέσματα φαίνονται στην παρακάτω εικόνα.



Σχήμα 5.4 Αποτελέσματα Εφαρμογής Data Augmentation

### 5.2.3 Εκπαίδευση Δικτύου

Το μοντέλο που επιλέχθηκε για την εκπαίδευση δέχεται για είσοδο μια εικόνα μεγέθους 32 x 32 x 1, ουσιαστικά μια τετράγωνη ασπρόμαυρη εικόνα και διαθέτει συνελικτικά επίπεδα που ακολουθούνται από επίπεδα μέγιστης ομαδοποίησης φίλτρων 2 x 2 για την εξαγωγή των χαρακτηριστικών. Τα χαρακτηριστικά αυτά, αφού τα μετατρέψουμε σε έναν μονοδιάστατο πίνακα, περνάνε σε ένα πλήρως συνδεδεμένο νευρωνικό δίκτυο 500 νευρώνων με πιθανότητα εγκατάλειψης 0.5 για την ταξινόμηση τους στις 43 κλάσεις.

Ως συνάρτηση ενεργοποίησης χρησιμοποιείται παντού η relu εκτός από τον τελευταίο κόμβο που χρησιμοποιείται η softmax για να εξάγουμε πιθανότητες. Ως συνάρτηση βελτιστοποίησης έχουμε την Adam με ρυθμό εκμάθησης 0,001 και ως συνάρτηση σφάλματος την sparse\_categorical\_crossentropy. Το summary του μοντέλου όπως εμφανίζεται, φαίνεται παρακάτω:

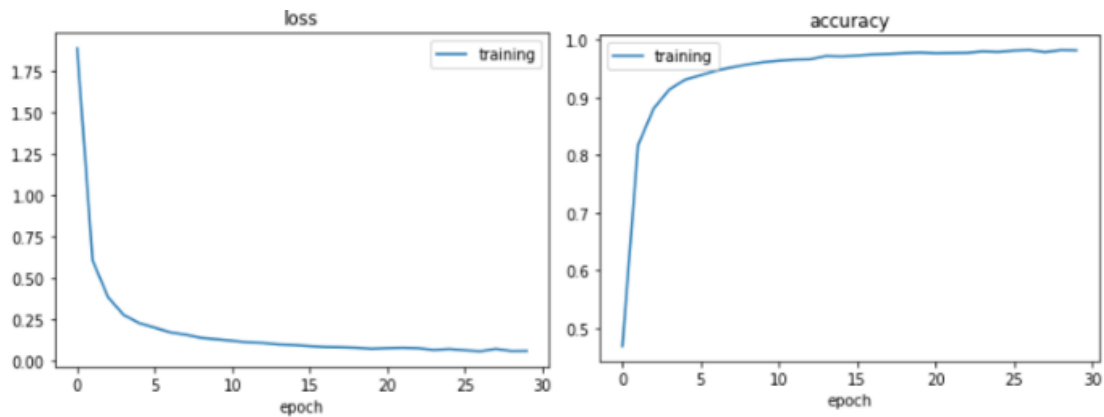
Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 60)	1560
conv2d_1 (Conv2D)	(None, 24, 24, 60)	90060
max_pooling2d (MaxPooling2D)	(None, 12, 12, 60)	0
conv2d_2 (Conv2D)	(None, 10, 10, 30)	16230
conv2d_3 (Conv2D)	(None, 8, 8, 30)	8130
max_pooling2d_1 (MaxPooling2D)	(None, 4, 4, 30)	0
flatten (Flatten)	(None, 480)	0
dense (Dense)	(None, 500)	240500
dropout (Dropout)	(None, 500)	0
dense_1 (Dense)	(None, 43)	21543
Total params: 378,023		
Trainable params: 378,023		
Non-trainable params: 0		

Σχήμα 5.5 Το Summary Του Μοντέλου Αναγνώρισης Των Πινακίδων

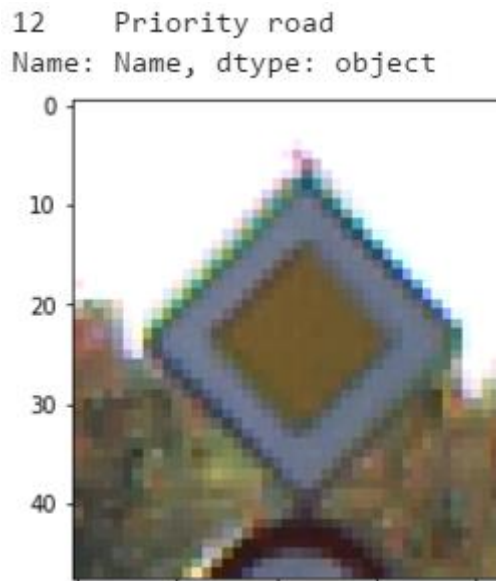
Όπως φαίνεται το μοντέλο μας έχει 378,023 παραμέτρους που πρέπει να εκπαιδευτούν, που είναι μικρός αριθμός. Η εκπαίδευση έγινε για 30 εποχές με batch size 50. Το batch size είναι ο αριθμός των εικόνων που τροφοδοτούνται στο δίκτυο κάθε φορά κατά τη διάρκεια της

εκπαίδευσης. Όσο μικρότερο το batch size τόσο καλύτερα αποτελέσματα έχει η εκπαίδευση αλλά αυξάνεται ο χρόνος που χρειάζεται. Το δίκτυο σε 30 εποχές κατάφερε να πετύχει ακρίβεια 98,15%.



Σχήμα 5.6 Αποτελέσματα Εκπαίδευσης

Πλέον, το μοντέλο μας είναι σε θέση να ταξινομεί τις πινακίδες στις 43 κλάσεις. Στο **σχήμα 5.9** φαίνεται ότι για μία εικόνα εισόδου που περιέχει μόνο την πινακίδα που θέλουμε να ταξινομήσουμε, το μοντέλο μας προβλέπει την κλάση στην οποία ανήκει, στην προκειμένη περίπτωση η δέκατη έβδομη (17) και στην συνέχεια μας δίνει και το όνομα αυτής, οδός προτεραιότητας (priority road).



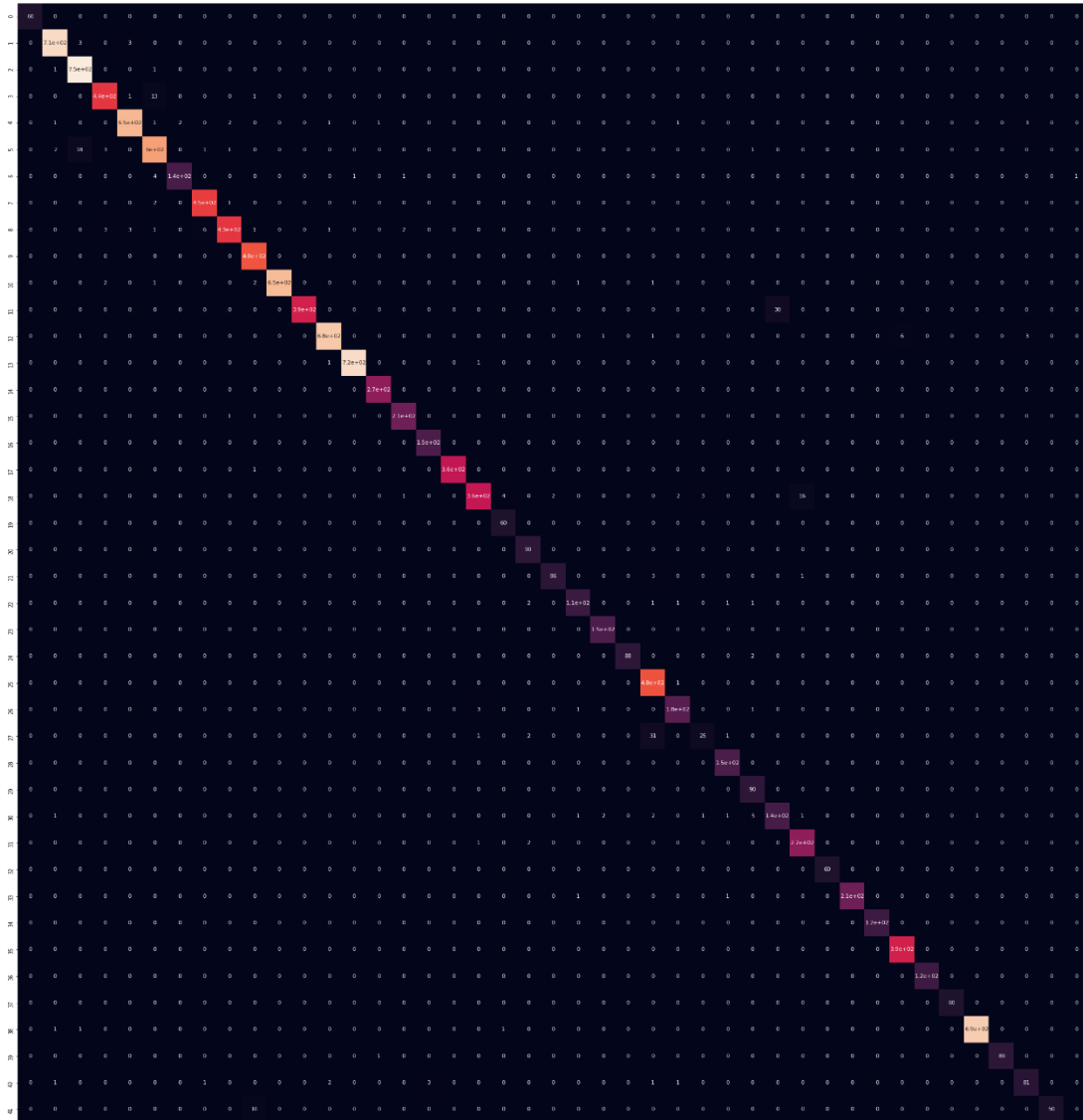
Σχήμα 5.7 Επιτυχημένη Πρόβλεψη Κλάσης Πινακίδας

#### 5.2.4 Αποτελέσματα ελέγχου

Μετά την εκπαίδευση του δικτύου, χρησιμοποιούμε τα δεδομένα ελέγχου για να μετρήσουμε και να οπτικοποιήσουμε τις προβλέψεις του μοντέλου μας σε νέα δεδομένα. Αρχικά, με χρήση της συνάρτησης `accuracy_score` της βιβλιοθήκης `scikit-learn` μετράμε την ακρίβεια με την οποία το μοντέλο μας αναγνωρίζει τις πινακίδες του συνόλου δεδομένου που προορίζονται για έλεγχο. Το μοντέλο κατάφερε να αναγνωρίσει τις νέες πινακίδες με 97,8 ακρίβεια που είναι μεγάλη επιτυχία.

Για καλύτερη απεικόνιση των αποτελεσμάτων ελέγχου χρησιμοποιούμε τη βιβλιοθήκη `seaborn`, ώστε να δημιουργήσουμε ένα `heatmap`. Το `heatmap` είναι μια τεχνική οπτικοποίησης δεδομένων που αποτυπώνει τη σημασία των δεδομένων με τη μορφή χρωμάτων. Στη δεδομένη περίπτωση στο Σχήμα 5.8 αποτυπώνεται ο `confusion matrix` του δικτύου. Ο `confusion matrix` είναι μια διάταξη πίνακα που μας επιτρέπει την απεικόνιση της επίδοσης του μοντέλου μας. Ο άξονας  $x$  συμβολίζει τις προβλέψεις του μοντέλου ενώ ο άξονας  $y$  τις πραγματικές τιμές. Το σημείο τομής των αξόνων μας δείχνει τις τιμές `true positives`, ενώ η οριζόντια σειρά που εκτείνεται από αυτό το σημείο μας δίνει τις `false negatives`. Η κάθετη σειρά που εκτείνεται από το σημείο τομής τους, δίνει τις `false positives`. Σε όλο το υπόλοιπο διάγραμμα βρίσκονται οι `true negatives`. Άρα η διαγώνια γραμμή στο `heatmap` μας δείχνει τις περιοχές που το μοντέλο μας έχει καλή απόδοση με τον αριθμό τιμών εκτός της διαγωνίου να είναι περιορισμένος.

`True positive (tp)` είναι η πρόβλεψη της οποίας η τιμή είναι αληθής την ώρα που η ετικέτα είναι επίσης αληθής. `True Negative (tn)` είναι η πρόβλεψη της οποίας η τιμή είναι ψευδής την ώρα που η ετικέτα είναι επίσης ψευδής. Ενώ αντιθέτως, `false negatives (fn)` και `false positives (fp)` είναι όταν η τιμή πρόβλεψης του μοντέλου είναι λανθασμένα ψευδής και αληθής αντίστοιχα.



Σχήμα 5.8 Ο Confusion Matrix Του Δικτύου

Πέρα από τον confusion matrix, η βιβλιοθήκη scikit-learn έχει ακόμα μία συνάρτηση για να ελέγχουμε τα αποτελέσματα ενός μοντέλου, την `classification_report`.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	60
1	0.99	0.99	0.99	720
2	0.97	1.00	0.98	750
3	0.98	0.97	0.97	450
4	0.99	0.98	0.99	660
5	0.96	0.96	0.96	630
6	0.98	0.95	0.97	150
7	0.98	0.99	0.99	450
8	0.99	0.96	0.98	450
9	0.97	1.00	0.98	480
10	1.00	0.99	0.99	660
11	1.00	0.93	0.96	420
12	0.99	0.98	0.99	690
13	1.00	1.00	1.00	720
14	0.99	1.00	1.00	270
15	0.98	0.99	0.99	210
16	0.98	1.00	0.99	150
17	1.00	1.00	1.00	360
18	0.98	0.93	0.96	390
19	0.92	1.00	0.96	60
20	0.94	1.00	0.97	90
21	0.98	0.96	0.97	90
22	0.97	0.95	0.96	120
23	0.98	1.00	0.99	150
24	1.00	0.98	0.99	90
25	0.92	1.00	0.96	480
26	0.97	0.97	0.97	180
27	0.86	0.42	0.56	60
28	0.97	1.00	0.99	150
29	0.90	1.00	0.95	90
30	0.82	0.90	0.86	150
31	0.94	0.99	0.96	270
32	1.00	1.00	1.00	60
33	1.00	0.99	1.00	210
34	1.00	1.00	1.00	120
35	0.98	1.00	0.99	390
36	1.00	1.00	1.00	120
37	1.00	1.00	1.00	60
38	1.00	1.00	1.00	690
39	1.00	0.99	0.99	90
40	0.91	0.90	0.91	90
41	1.00	0.83	0.91	60
42	0.99	0.93	0.96	90
accuracy			0.98	12630
macro avg	0.97	0.96	0.97	12630
weighted avg	0.98	0.98	0.98	12630

Σχήμα 5.9 To Classification\_Report Του Δικτύου

Η μετρική precision μας δείχνει πόσο ακριβές είναι το μοντέλο μας σε κάθε κλάση. Πρακτικά συγκρίνει πόσες προβλέψεις είναι ολόσωστες, από τις συνολικές. Μαθηματικά αυτό ορίζεται ως

$$\text{precision} = \frac{tp}{tp + fp} \quad (5.1)$$

Η μετρική Recall μας δείχνει την ικανότητα του μοντέλου να βρει όλα τα σημεία δεδομένων που μας ενδιαφέρουν σε ένα σύνολο δεδομένων.

$$\text{Recall} = \frac{tp}{tp + fn} \quad (5.2)$$

Η βαθμολογία f1-score μας δίνει τον αρμονικό μέσο των προηγούμενων δύο μετρικών,

$$F_1 = \left( \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} \right) = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (5.3)$$

,ενώ η μετρική support είναι ο αριθμός δειγμάτων που βρίσκονται στην κάθε κλάση.

Η ακρίβεια (accuracy) είναι το άθροισμα των true positives και true negatives δια του συνολικού αριθμού δειγμάτων

$$\text{accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \quad (5.3)$$

Αυτός ο τρόπος υπολογισμού της ακρίβειας είναι ακριβές μόνο εάν το μοντέλο είναι ισορροπημένο. Σε αντίθετη περίπτωση ανισορροπίας των κλάσεων δεν θα δώσει καλά αποτελέσματα.

### 5.3 ΕΝΤΟΠΙΣΜΟΣ ΚΥΚΛΟΦΟΡΙΑΚΗΣ ΣΗΜΑΝΣΗΣ

Όπως είδαμε σε προηγούμενο κεφάλαιο, στον εντοπισμό αντικειμένων, το δίκτυο δέχεται ως είσοδο μια εικόνα και προσπαθεί όχι μόνο να ταξινομήσει τα αντικείμενα σε κλάσεις αλλά να βρει και τις ακριβείς συντεταγμένες τους στην εικόνα. Αυτό θα ήταν αρκετό για να λύσουμε το πρόβλημα της αναγνώρισης των πινακίδων αλλά τα σύνολα δεδομένων που έχουμε στη διάθεσή μας δεν έχουν αρκετά μεγάλο πλήθος εικόνων εκπαίδευσης ώστε να επιτευχθεί αυτό. Γι' αυτό, θα εκπαιδεύσουμε ένα δίκτυο να εντοπίζει τις πινακίδες κυκλοφοριακής σήμανσης χωρίς να τις ταξινομεί σε κλάσεις και στη συνέχεια θα τροφοδοτούμε τις πινακίδες που εντοπίσαμε στο νευρωνικό δίκτυο της προηγούμενης ενότητας το οποίο θα τις ταξινομεί.

### 5.3.1 Σύνολο Δεδομένων (Dataset)

Το σύνολο δεδομένων που επιλέχθηκε για την εκπαίδευση ενός δικτύου που επιτυγχάνει τον εντοπισμό των πινακίδων είναι το και είναι το μόνο ελεύθερα διαθέσιμο σύνολο δεδομένων για αυτόν τον σκοπό που περιέχει πινακίδες ευρωπαϊκού τύπου. Περιέχει 852 εικόνες δρόμου που τραβήχτηκαν μέσα από αυτοκίνητο την ώρα της οδήγησης διαστάσεων (1360 x 800) και περιέχουν τουλάχιστον μία πινακίδα κυκλοφοριακής σήμανσης από τις 43 κλάσεις πάνω στις οποίες εκπαιδεύσαμε το μοντέλο αναγνώρισης της κυκλοφοριακής σήμανσης στην προηγούμενη ενότητα.



Σχήμα 5.10 Παραδείγματα Εικόνων Του Συνόλου Δεδομένων Εντοπισμού Κυκλοφοριακής Σήμανσης

Επίσης, διαθέτει ένα .txt της μορφής

00000.ppm;774;411;815;446;11

Σχήμα 5.11 Παράδειγμα Αρχείου .txt Του Συνόλου Δεδομένων Εντοπισμού Κυκλοφοριακής Σήμανσης

στο οποίο περιέχονται όλες οι απαραίτητες πληροφορίες για τον εντοπισμό και την κατηγοριοποίηση της κάθε πινακίδας. Για παράδειγμα η πάνω εικόνα μας λέει πως το αρχείο με όνομα 00000.ppm περιέχει μια πινακίδα της ενδέκατης κατηγορίας στην θέση που περιγράφουν



οι 4 άλλοι αριθμοί από τους οποίους ο πρώτος είναι το πάνω αριστερό σημείο του πλαισίου στον άξονα των x (x\_top) ο δεύτερος στον άξονα των y (y\_top) ο τρίτος το πλάτος του πλαισίου (width) και ο τέταρτος το ύψος (height).

Τέλος περιέχει ένα ακόμα .csv αρχείο που εξηγεί τι αντιπροσωπεύει η κάθε κατηγορία πινακίδας

ClassId	Name
0	Speed limit (20km/h)
1	Speed limit (30km/h)
2	Speed limit (50km/h)
3	Speed limit (60km/h)
4	Speed limit (70km/h)
5	Speed limit (80km/h)
6	End of speed limit (80km/h)
7	Speed limit (100km/h)
8	Speed limit (120km/h)
9	No passing
10	No passing for vechiles over 3.5 metric tons
11	Right-of-way at the next intersection
12	Priority road
13	Yield
14	Stop
15	No vechiles
16	Vechiles over 3.5 metric tons prohibited
17	No entry
18	General caution
19	Dangerous curve to the left
20	Dangerous curve to the right
21	Double curve
22	Bumpy road
23	Slippery road
24	Road narrows on the right
25	Road work
26	Traffic signals
27	Pedestrians
28	Children crossing
29	Bicycles crossing
30	Beware of ice/snow
31	Wild animals crossing
32	End of all speed and passing limits
33	Turn right ahead
34	Turn left ahead
35	Ahead only
36	Go straight or right
37	Go straight or left
38	Keep right
39	Keep left
40	Roundabout mandatory
41	End of no passing

Σχήμα 5.12 Οι 43 κλάσεις του Συνόλου Δεδομένων Εντοπισμού Κυκλοφοριακής Σήμανσης

Οι 43 κλάσεις στο σύνολο δεδομένων κατανέμονται σε 4 ευρύτερες κλάσεις (prohibitory, danger, mandatory, other) που διαχωρίζει τις πινακίδες με βάση τη πληροφορία που παρέχουν αλλά και το σχήμα και το χρώμα. Συγκεκριμένα ο διαχωρισμός φαίνεται στο Σχήμα 5.13 Διαχωρισμός Των Κλάσεων του Συνόλου Δεδομένων Εντοπισμού Κυκλοφοριακής Σήμανσης Σε Υπό-κλάσεις.



Σχήμα 5.13 Διαχωρισμός Των Κλάσεων του Συνόλου Δεδομένων Εντοπισμού Κυκλοφοριακής Σήμανσης Σε Υπό-κλάσεις

### 5.3.2 Προ-Επεξεργασία Δεδομένων

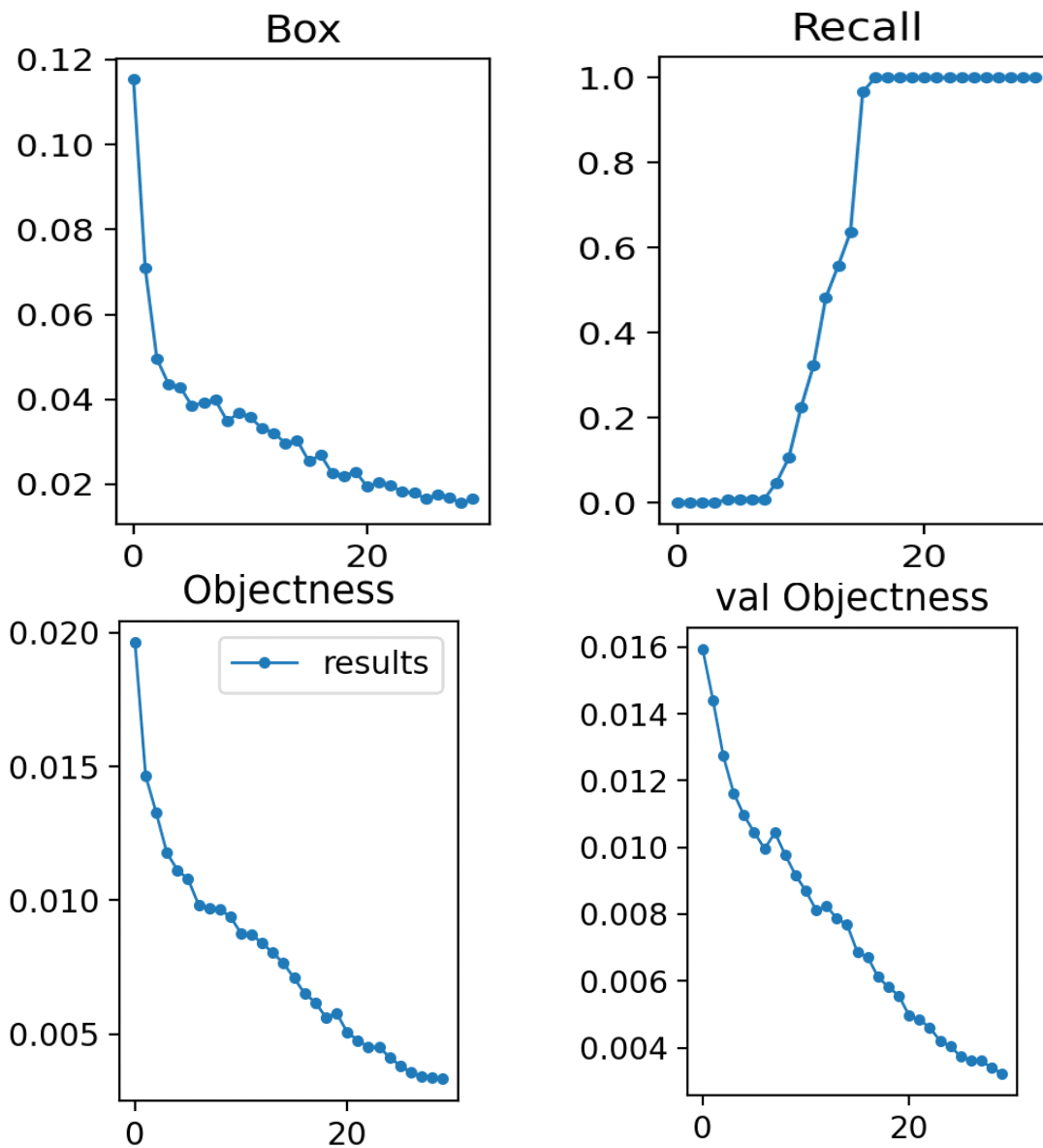
Η αρχική ιδέα ήταν να εκπαιδευτεί το δίκτυο σε κάθε μια κατηγορία πινακίδων ξεχωριστά έτσι ώστε όχι μόνο να τις εντοπίζει σαν πινακίδες αλλά να τις αναγνωρίζει κιόλας. Ωστόσο, το σύνολο δεδομένων είναι πάρα πολύ μικρό για να γίνει αυτό καθώς για κάποιες κλάσεις διαθέτουμε λιγότερες από 20 εικόνες. Έπειτα, έγινε προσπάθεια εκπαίδευσης για τις 4 ευρύτερες κλάσεις μιας και έτσι θα έχουμε περισσότερα δεδομένα για την καθεμία. Όμως, ούτε αυτό είχε καλά αποτελέσματα.

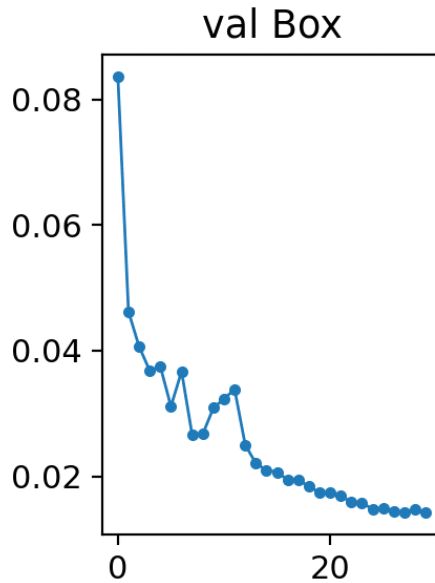
Οπότε, τροποποιήσαμε το .txt αρχείο αλλάζοντας όλες τις κατηγορίες σε 0. Με αυτόν τον τρόπο, δεν θα μπορεί να αναγνωρίζει τις πινακίδες αλλά θα έχει αρκετά δεδομένα ώστε να τις εντοπίζει με αρκετά μεγάλη ακρίβεια. Στη συνέχεια, χωρίσαμε το σύνολο δεδομένων σε 800 εικόνες για εκπαίδευση και τις υπόλοιπες για επικύρωση. Τέλος, φέρνουμε τα δεδομένα μας σε μορφή

αποδεκτή από το YOLO v5, τα ταξινομούμε κατάλληλα σε φακέλους και προχωράμε στην εκπαίδευση του δικτύου.

### 5.3.3 Εκπαίδευση Δικτύου

Για την εκπαίδευση επιλέχθηκε ο αλγόριθμος YOLO v5 που αποτελεί την πιο πρόσφατη έκδοση YOLO ως σήμερα. Η κύρια βελτίωση σε σχέση με τις προηγούμενες εκδόσεις βρίσκεται στην τεχνική data augmentation χρησιμοποιείται για να αυξήσει τα δεδομένα που τροφοδοτούνται στο δίκτυο. Η εκπαίδευση έγινε για 30 εποχές με batch size 8. Τα αποτελέσματα της εκπαίδευσης φαίνονται σχηματικά στις παρακάτω εικόνες.



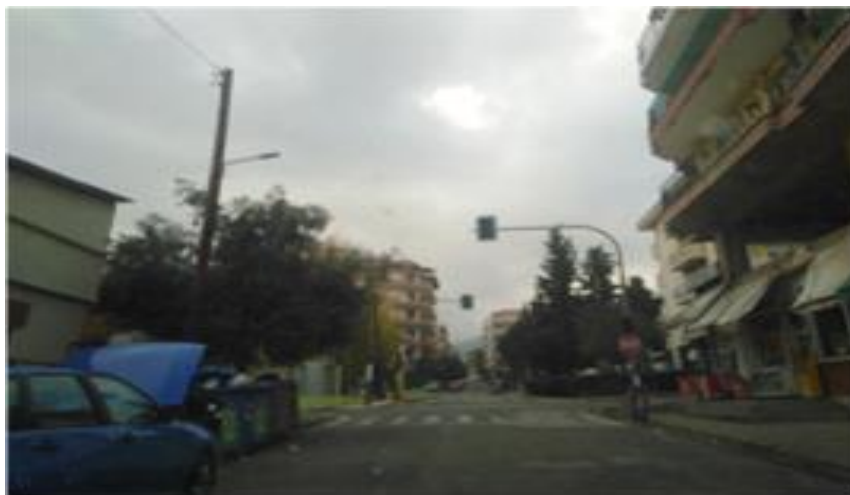


Σχήμα 5.14 Αποτελέσματα εκπαίδευσης δικτύου

#### 5.4 ΣΥΝΔΥΑΣΜΟΣ ΤΩΝ ΔΙΚΤΥΩΝ

Πλέον, εύκολα μπορεί να αντιληφθεί κανείς πως προκειμένου να αναγνωρίσουμε επιτυχώς πινακίδες από εικόνες δρόμου την ώρα της οδήγησης, θα πρέπει να συνδυάσουμε τα δίκτυα των προηγούμενων δύο ενότητων.

Αρχικά, τροφοδοτούμε την εικόνα πάνω στην οποία θέλουμε να εντοπίσουμε τις πινακίδες στο μοντέλο YOLO που έχουμε εκπαιδεύσει να εντοπίζει τις πινακίδες και αυτό μας επιστρέφει την εικόνα μας στην οποία έχει προσθέσει ορθογώνια σε κάθε πινακίδα που έχει εντοπίσει και από πάνω την κατηγορία της (στην δικιά μας περίπτωση οι κατηγορία είναι μόνο μια).





Σχήμα 5.15 Εντοπισμός Πινακίδων Σε Εικόνα

Επίσης, μας επιστρέφει ένα .txt αρχείο που μας δίνει πληροφορίες για την κατηγορία και την τοποθεσία στην οποία τις εντόπισε.

```
testimage.txt X
1 0 0.727764 0.684936 0.0266827 0.0352564
2
```

Σχήμα 5.16 Παραγόμενο Αρχείο .txt Από Εντοπισμό

Στην συνέχεια, αξιοποιώντας τις πληροφορίες από το .txt αρχείο, κόβουμε από την αρχική εικόνα τα σήματα που εντοπίσαμε και τα αποθηκεύουμε ως ξεχωριστές εικόνες.

“File saved successfully”



Αυτές τις εικόνες τις προωθούμε στο νευρωνικό δίκτυο το οποίο εκπαιδεύσαμε στην ενότητα 5.2 ώστε να αναγνωρίζει τις πινακίδες και αυτό με την σειρά του μας επιστρέφει την κατηγορία της κάθε πινακίδας.

Έτσι, με τη βοήθεια των δύο νευρωνικών δικτύων καταφέραμε να εντοπίσουμε και να αναγνωρίσουμε τις πινακίδες κυκλοφοριακής σήμανσης σε μια εικόνα τραβηγμένη την ώρα της οδήγησης παρόλο που δεν είχαμε στη διάθεσή μας ένα σύνολο δεδομένων με αρκετές εικόνες για την εκπαίδευση ενός μοντέλου εντοπισμού.

*Σχήμα 5.17 Αποθήκευση Εικόνας Πινακίδας*

## 5.5 ΑΝΑΓΝΩΡΙΣΗ ΦΩΤΕΙΝΟΥ ΣΗΜΑΤΟΔΟΤΗ

Μία από τις κατηγορίες που εντοπίζει το προ-εκπαιδευμένο μοντέλο του YOLO πάνω στο σετ δεδομένων COCO που χρησιμοποιήσαμε στην ενότητα 4.3 είναι και του φωτεινού σηματοδότη.

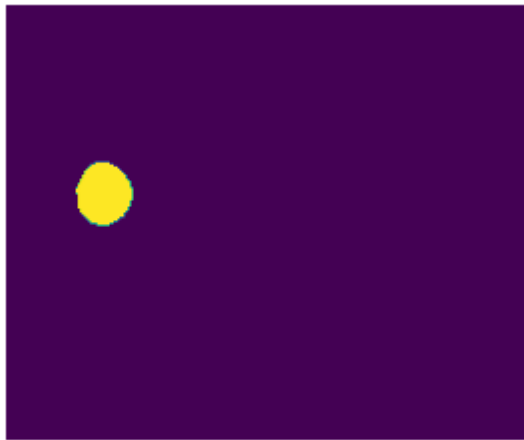
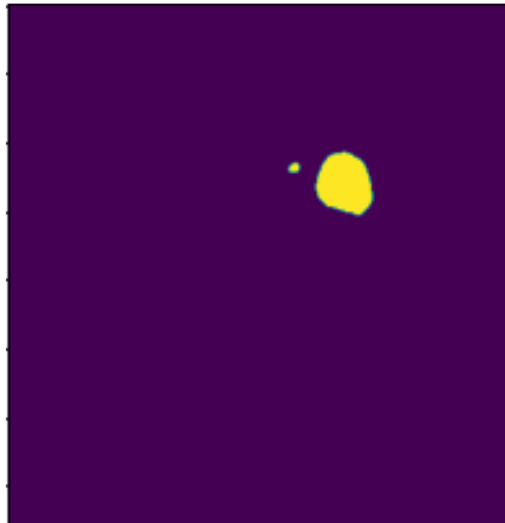
```
14 Stop  
Name: Name, dtype: object  
Probability: 98.99075031280518
```



*Σχήμα 5.18 Αναγνώριση Πινακίδας*

Ωστόσο, δεν μπορεί να αναγνωρίσει εάν είναι πράσινο, πορτοκαλί ή κόκκινο.

Ένας απλός και πολύ εύκολος τρόπος να το πετύχουμε αυτό είναι, αφού πρώτα κόψουμε και αποθηκεύσουμε από την συνολική εικόνα μόνο το τμήμα του φωτεινού σηματοδότη, αρχικά να περάσουμε την εικόνα από "μάσκες" οι οποίες να αφήνουν πρακτικά να περνάνε μόνο τα χρώματα που μας ενδιαφέρουν. Θα χρειαστούμε δηλαδή 3 μάσκες, μία για το πράσινο, μία για το πορτοκαλί και μία για το κόκκινο αντίστοιχα.







Σχήμα 5.19 Παραδείγματα Εφαρμογής Χρωματικών Φίλτρων Σε Εικόνες Φωτεινών Σηματοδοτών

Έπειτα, επειδή καμιά φορά το φόντο της εικόνας, ακόμα και ο ίδιος ο φωτεινός σηματοδότης μπορεί να περιέχει ένα ή περισσότερα από τα χρώματα που περνάνε από τις μάσκες μας και αυτό μπορεί να οδηγήσει σε λανθασμένη εκτίμηση εάν χρησιμοποιήσουμε μόνο το χρώμα, με την βοήθεια της συνάρτησης `HoughCircles` του `OpenCV`, θα εντοπίσουμε τους κύκλους που εμφανίζονται στην εικόνα μετά την κάθε μάσκα. Έτσι, σε όποια μάσκα εμφανιστεί κύκλος, ο φωτεινός σηματοδότης έχει το χρώμα αυτό.

red



green





orange



Σχήμα 5.20 Πρόβλεψη Χρώματος Φωτεινών Σηματοδοτών

Στην παραπάνω εικόνα φαίνονται τα αποτελέσματα της αναγνώρισης των φωτεινών σηματοδοτών μετά την εφαρμογή των масκών και της συνάρτησης εύρεσης κύκλων του OpenCV.

Παρόλα αυτά, στην πραγματικότητα, όλοι οι σηματοδότες δεν έχουν κυκλικό σχήμα, υπάρχουν και σηματοδότες που απεικονίζονται με βέλη. Αυτό θα μπορούσε να αποτελεί πρόβλημα στον συγκεκριμένο τρόπο επίλυσης. Όμως, μπορούμε να τροποποιήσουμε τον αλγόριθμό μας αλλάζοντας τις παραμέτρους της συνάρτησης HoughCircles του OpenCV ώστε να βρίσκει και λίγο μικρότερους κύκλους εντοπίζοντας λιγότερα σημεία. Με αυτόν το τρόπο, εντοπίζει κύκλο στην κορυφή του βέλους.

green  
orange  
orange



Σχήμα 5.21 Πρόβλεψη Διαφόρων Σχημάτων  
Φωτεινών Σηματοδοτών

Στην παραπάνω εικόνα φαίνεται ότι εντοπίσαμε 2 πορτοκαλί κύκλους στα βέλη και έναν πράσινο κύκλο στον κυκλικό φωτεινό σηματοδότη.

## ΚΕΦΑΛΑΙΟ 6ο

### Εύρεση επιτρεπτής περιοχής οδήγησης

#### 6.1 ΕΙΣΑΓΩΓΗ

Δεν είναι δυνατό να αναφερόμαστε σε αυτόνομη οδήγηση εάν το όχημα μας δεν είναι σε θέση να αναγνωρίζει την περιοχή στην οποία μπορεί να οδηγήσει. Πλέον, τα περισσότερα αν όχι όλα τα σύγχρονα αυτοκίνητα που πωλούνται στην αγορά διαθέτουν συστήματα τα οποία βοηθούν τον οδηγό να διατηρήσει το όχημα του στο μέσο της λωρίδας οδήγησης.

Σε δρόμους με επαρκή διαγράμμιση και με καλές καιρικές συνθήκες, η διαδικασία της εύρεσης της λωρίδας οδήγησης μπορεί να πραγματοποιηθεί με αρκετά απλές μεθόδους χωρίς την χρήση νευρωνικών δικτύων και βαθιάς μάθησης. Ωστόσο, για δρόμους με ανεπαρκή ή ακόμα και ελλιπή διαγράμμιση θα πρέπει το όχημά μας να εντοπίζει με ακρίβεια την επιτρεπτή περιοχή που μπορεί να κατευθυνθεί. Εδώ, προκειμένου να επιτευχθούν υψηλά ποσοστά ακρίβειας είναι απαραίτητη η ανάπτυξη ενός συνελκτικού νευρωνικού δικτύου ικανό να διαχωρίζει σε μια εικόνα δρόμου την επιτρεπτή περιοχή οδήγησης από περιοχές στις οποίες δεν επιτρέπεται να οδηγήσει το όχημά μας, όπως για παράδειγμα τα πεζοδρόμια.

#### 6.2 ΔΙΑΤΗΡΗΣΗ ΤΟΥ ΟΧΗΜΑΤΟΣ ΣΤΟ ΜΕΣΟ ΤΗΣ ΛΩΡΙΔΑΣ ΟΔΗΓΗΣΗΣ

Σε αυτήν την ενότητα θα αναπτυχθεί η εύρεση της λωρίδας οδήγησης σε δρόμους με επαρκή διαγράμμιση ως πρόβλημα μηχανικής όρασης χωρίς τη χρήση βαθιάς μάθησης. Ωστόσο, για ένα αυτοκινούμενο όχημα δεν είναι αρκετή η εύρεση της λωρίδας που επιτρέπεται να κινηθεί αλλά θα πρέπει να είναι σε θέση να μπορεί να διατηρηθεί στο μέσο αυτής κατά τη διάρκεια της οδήγησης.

##### 6.2.1 Ανίχνευση Των Γραμμών Της Λωρίδας Οδήγησης

Εφόσον υπάρχει διαγράμμιση στο δρόμο και είναι εμφανή, η διαδικασία της αναγνώρισης της λωρίδας οδήγησης που θα πρέπει να παραμείνει το όχημά μας μπορεί να πραγματοποιηθεί απλά με την ανίχνευση των γραμμών, αφού αυτές είναι που την καθορίζουν.

Αρχικά, στην εικόνα δρόμου μας με τη βοήθεια της συνάρτησης Canny της βιβλιοθήκης του OpenCV, γίνεται η ανίχνευση των ακμών που βρίσκονται μέσα σε αυτήν. Η τεχνική της ανίχνευσης των ακμών χρησιμοποιείται σε διάφορα συστήματα όρασης υπολογιστών καθώς

συμβάλει στην εξαγωγή χρήσιμων δομικών πληροφοριών της εικόνας και τη μεγάλη μείωση του όγκου δεδομένων προς επεξεργασία.



Σχήμα 6.1 Εφαρμογή Της Συνάρτησης Canny Στην Εικόνα Εισόδου

Στην συνέχεια προκειμένου να μειώσουμε ακόμα περισσότερο τα δεδομένα που πρέπει να επεξεργαστεί το πρόγραμμά μας εφαρμόζουμε μια "μάσκα" η οποία περιορίζει την εικόνα μόνο στην περιοχή που μας ενδιαφέρει, δηλαδή την περιοχή στην οποία θα πρέπει θεωρητικά να βρίσκονται οι γραμμές της λωρίδας οδήγησης. Η "μάσκα" αυτή είναι ουσιαστικά μία εικόνα ίδιων διαστάσεων με την εικόνα εισόδου και το κάθε εικονοστοιχείο που δεν περιλαμβάνεται στην περιοχή που μας ενδιαφέρει έχει την τιμή μηδέν ενώ τα υπόλοιπα έχουν την τιμή ένα.



Σχήμα 6.2 Μάσκα Περιοχής Ενδιαφέροντος

Με αυτόν τον τρόπο όταν το κάθε εικονοστοιχείο της εικόνας αυτής πολλαπλασιαστεί με το αντίστοιχο της εικόνας ακμών, το αποτέλεσμα θα είναι μια νέα εικόνα στην οποία η περιοχή που μας ενδιαφέρει έχει μείνει άθικτη καθώς η τιμή των εικονοστοιχείων έχει πολλαπλασιαστεί με το ένα, ενώ ολόκληρη η υπόλοιπη εικόνα είναι μαύρη αφού η τιμή των εικονοστοιχείων της θα είναι μηδέν.



Σχήμα 6.3 Εικόνα Ακμών Της Περιοχής Ενδιαφέροντος

Έπειτα, με τη βοήθεια της συνάρτησης `HoughLinesP` του `OpenCV`, θα εντοπίσουμε τα ευθύγραμμα τμήματα που εντοπίζονται στην εικόνα μας. Η συνάρτηση αυτή μας επιστρέφει τις συντεταγμένες του αρχικού και τελικού σημείου του κάθε ευθυγράμμου τμήματος. Η κάθε γραμμή της λωρίδας οδήγησης μπορεί να αποτελείται από περισσότερα από ένα ευθύγραμμα τμήματα καθώς η διαχωριστική γραμμή στο δρόμο μπορεί να είναι διακεκομμένη ή διπλή. Επίσης, όπως φαίνεται στο Σχήμα 6.4, την κάθε γραμμή μπορεί να την αναγνωρίζει σαν δύο, λόγω του πάχους της.



Σχήμα 6.4 Εύρεση Γραμμών

Τέλος, για καλύτερη απεικόνιση και για πιο εύκολη εξαγωγή πληροφοριών, χρησιμοποιώντας τις συντεταγμένες που πήραμε από την εφαρμογή της συνάρτησης `HoughLinesP` βρίσκουμε τον μέσο όρο των κλίσεων των ευθυγράμμων τμημάτων με αρνητική τιμή κλίσης και τον μέσο όρο των κλίσεων των ευθυγράμμων τμημάτων με θετική τιμή κλίσης, όπως και το μέσο σημείο που τα ευθύγραμμα τμήματα αυτά τέμνουν τον άξονα των  $x$ . Τα ευθύγραμμα τμήματα με θετική τιμή κλίσης αντιπροσωπεύουν την αριστερή γραμμή της λωρίδας οδήγησης ενώ αυτά με αρνητική τη

δεξιά. Έτσι, έχοντας την κλίση και το μέσο σημείο που τέμνει η κάθε γραμμή τον άξονα  $x$ , σχηματίζουμε τις γραμμές αυτές στην τελική εικόνα και τις προεκτείνουμε ως το μέσο της.

### 6.2.2 Συμβουλές Πλοήγησης

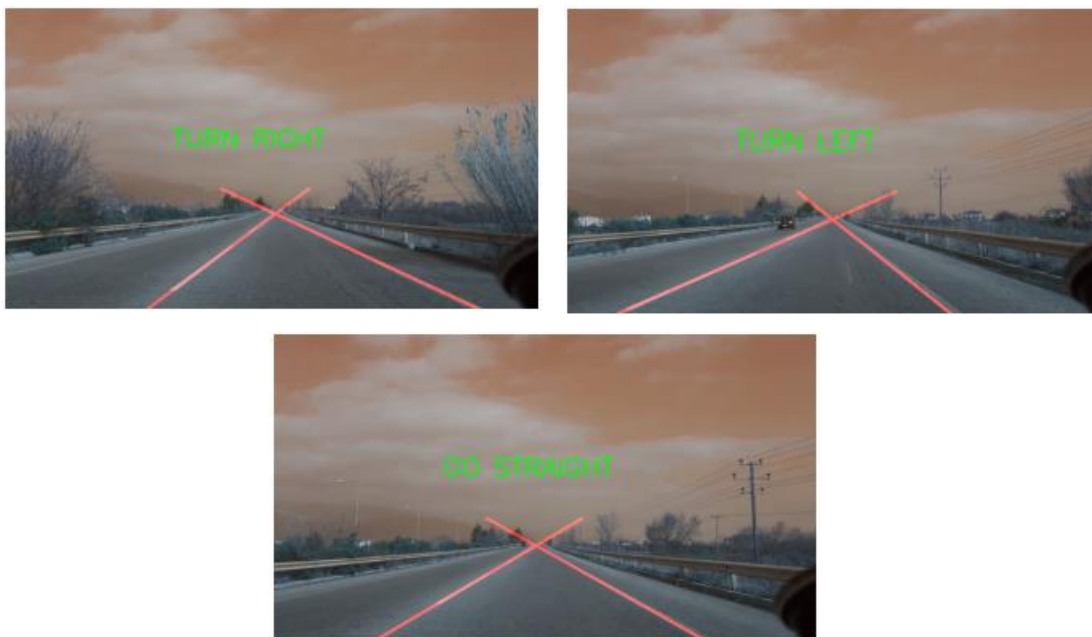


Σχήμα 6.5 Επέκταση Γραμμών



Πλέον, το πρόγραμμά μας είναι σε θέση να αναγνωρίζει, σε δρόμους με επαρκή διαγράμμιση, τα όρια της λωρίδας που επιτρέπεται να οδηγήσει. Το μόνο που μένει να κάνει είναι να παρέχει τις κατάλληλες συμβουλές είτε στον οδηγό είτε απευθείας στο τιμόνι έτσι ώστε το όχημά μας να παραμένει μέσα στα όρια αυτά.

Θα πρέπει πριν ξεκινήσουμε οποιαδήποτε διαδικασία να εντοπίσουμε το ακριβές πλάτος της εικόνας που αντιπροσωπεύει το κέντρο του οχήματος. Στην περίπτωση του παραδείγματός μας, η κάμερα είναι τοποθετημένη στο κέντρο του οχήματος, οπότε το μέσο της εικόνας αντιπροσωπεύει το κέντρο του οχήματος. Τώρα, εφόσον έχουμε τα ακριβή σημεία στα οποία η κάθε λωρίδα του δρόμου τέμνει τον άξονα των  $x$ , βρίσκουμε το μέσο αυτών και αν η τιμή του είναι μεγαλύτερη από το μισό πλάτος της εικόνας τότε το πρόγραμμά μας δίνει συμβουλή στον οδηγό ή το τιμόνι να στρίψει δεξιά (turn right), αν είναι μικρότερη να στρίψει αριστερά (turn left) και αν είναι ίσα να συνεχίσει ευθεία (go straight).



Σχήμα 6.6 Παραδείγματα Συμβουλών πλοήγησης Κατά Την Εύρεση Δύο Γραμμών

Σε αρκετούς δρόμους παρατηρούμε πως υπάρχει διαγράμμιση μόνο από τη μία πλευρά της λωρίδας, είτε αυτή είναι η διαχωριστική είτε η δεξιά που οριοθετεί τη λωρίδα. Προκειμένου το πρόγραμμά μας να δίνει συμβουλές ακόμα και όταν εντοπίζει τη μια από τις δύο γραμμές θα πρέπει να γνωρίζουμε το πλάτος της λωρίδας στην εικόνα μας. Έτσι, παίρνοντας μια εικόνα στην οποία εντοπίζονται και οι δύο γραμμές της λωρίδας, βρίσκουμε το πλάτος της. Το πλάτος της λωρίδας οδήγησης είναι σταθερό σε όλους τους δρόμους. Οπότε, αφού γνωρίζουμε την κλίση και το σημείο τομής της μιας γραμμής, από την κλίση της καταλαβαίνουμε αν αντιπροσωπεύει το αριστερό ή το δεξί όριο της λωρίδας και αντίστοιχα προσθέτουμε ή αφαιρούμε το πλάτος της λωρίδας από το σημείο τομής της γραμμής με τον άξονα των  $x$ . Με αυτόν τον τρόπο γνωρίζουμε τα σημεία τομής των γραμμών που οριοθετούν τη λωρίδα οδήγησης και ακολουθούμε την ίδια διαδικασία με πριν.



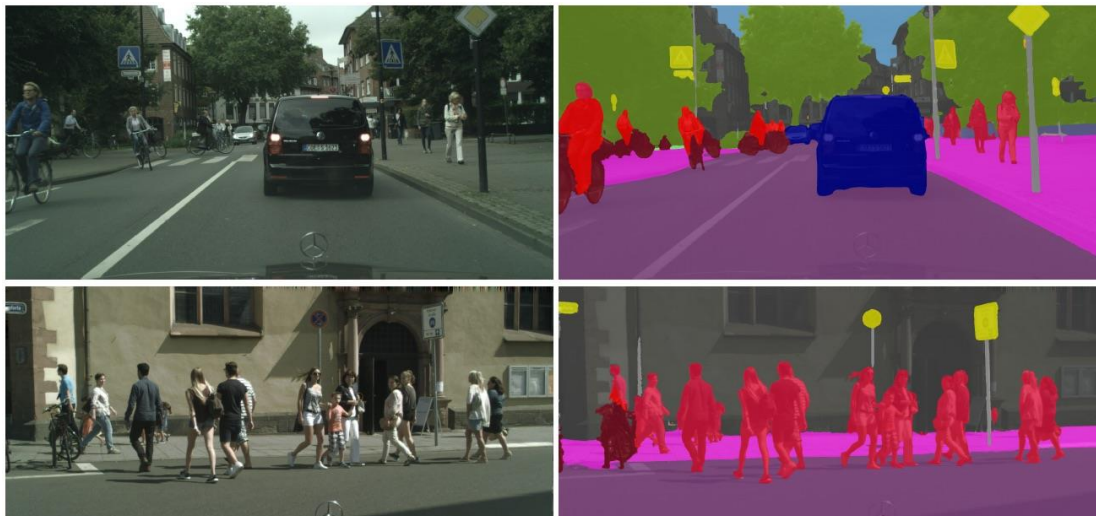
Σχήμα 6.7 Παραδείγματα Συμβουλών Πλοήγησης Κατά Την Εύρεση Μίας Γραμμής

Όπως εύκολα μπορεί να αντιληφθεί κανείς, η τεχνική αυτή έχει αρκετά καλά αποτελέσματα σε ευθείς δρόμους με επαρκή διαγράμμιση. Ωστόσο, σε στροφές επειδή βρίσκουμε τον μέσο όρο των κλίσεων των ευθυγράμμων τμημάτων και το μέσο σημείο που θα έτεμναν θεωρητικά τον άξονα των  $x$ , η ακρίβεια μειώνεται δραματικά. Μία λύση σε αυτό θα ήταν να αλλάξουμε την περιοχή που μας ενδιαφέρει η πρόβλεψη, αλλάζοντας τη μάσκα, σε πολύ πιο κοντινή απόσταση από το όχημα ώστε να μειώσουμε το σφάλμα από τους μέσους όρους.

### 6.3 ΤΜΗΜΑΤΙΚΗ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ (SEMANTIC SEGMENTATION)

Η τμηματική κατηγοριοποίηση είναι μια εργασία όρασης του υπολογιστή, κατά την οποία κατηγοριοποιούνται συγκεκριμένες περιοχές της εικόνας σύμφωνα με αυτό που εμφανίζεται. Πιο συγκεκριμένα, ο στόχος της τμηματικής κατηγοριοποίησης μιας εικόνας είναι η αντιστοίχιση του κάθε εικονοστοιχείου της με την κατηγορία που αντιπροσωπεύει. Η τεχνική αυτή συναντάται συχνά σε αρκετές εφαρμογές αυτόνομης οδήγησης και ιατρικής.

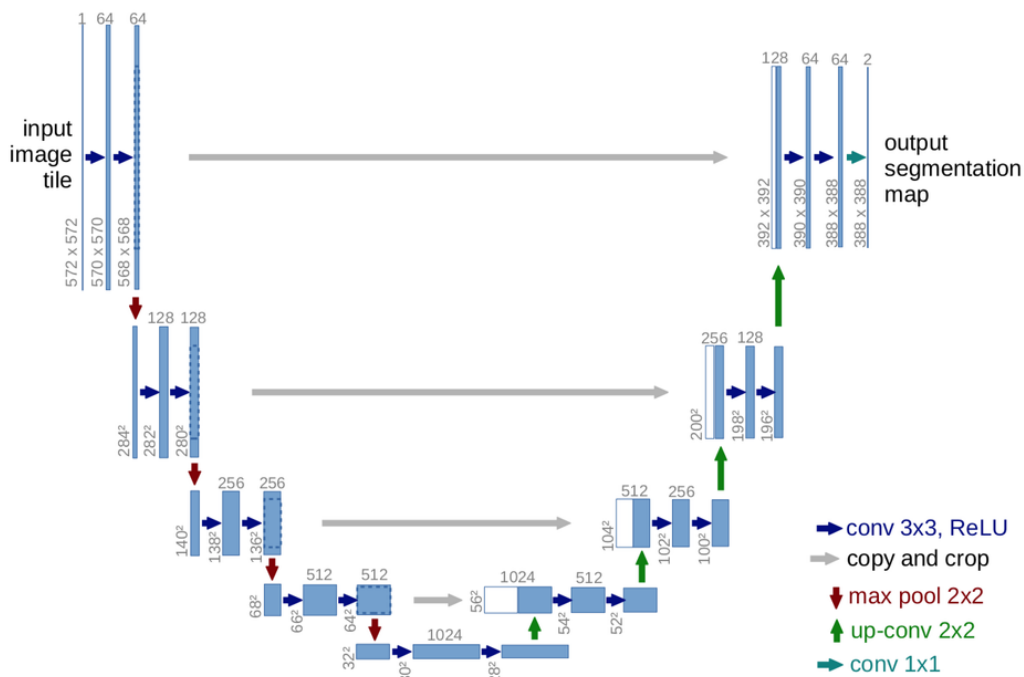




Σχήμα 6.8 Τμηματική Κατηγοριοποίηση Σε Εικόνες Δρόμου [33]

Όπως φαίνεται στην παραπάνω εικόνα, η διαδικασία της τμηματικής κατηγοριοποίησης είναι ακόμα πιο πολύπλοκη από αυτήν της ταξινόμησης και του εντοπισμού των διαφόρων αντικειμένων καθώς θα πρέπει, πέρα από την ταξινόμηση και τον εντοπισμό, το δίκτυο να μάθει με ακρίβεια το ακριβές όριο των αντικειμένων στην εικόνα.

Η πιο δημοφιλής αρχιτεκτονική δικτύου επίλυσης προβλημάτων τμηματικής κατηγοριοποίησης είναι η Unet [20]. Έχει πάρει την ονομασία από το αγγλικό γράμμα U καθώς η σχηματική απεικόνισή του μοιάζει με αυτό, όπως φαίνεται και στο παρακάτω σχήμα. Αποτελείται από έναν κωδικοποιητή που κάνει υποδειγματοληψία στην εικόνα εισόδου σε έναν χάρτη χαρακτηριστικών και από έναν αποκωδικοποιητή που δειγματίζει προς τα πάνω το χάρτη χαρακτηριστικών για να εισάγει το μέγεθος της εικόνας χρησιμοποιώντας τα στρώματα αποσυμπίεσης.



Σχήμα 6.9 Σχηματική Απεικόνιση Του Δικτύου UNET

## 6.4 ΕΥΡΕΣΗ ΕΠΙΤΡΕΠΤΗΣ ΠΕΡΙΟΧΗΣ ΟΔΗΓΗΣΗΣ

Η τεχνική της ανίχνευσης των γραμμών μας παρέχει μια εκτίμηση για την λωρίδα οδήγησης, ωστόσο δεν μπορεί να θεωρηθεί αρκετή. Για καλύτερα αποτελέσματα και για εκτιμήσεις ακόμα και σε δρόμους χωρίς διαγράμμιση είναι απαραίτητη η χρήση της βαθιάς μάθησης και η ανάπτυξη ενός δικτύου τμηματικής κατηγοριοποίησης του δρόμου.

### 6.4.1 Σύνολο Δεδομένων (Dataset)

Το σύνολο δεδομένων που επιλέχθηκε για την εκπαίδευση του δικτύου τμηματικής κατηγοριοποίησης του δρόμου είναι το BDD100K (Berkeley Deep Drive). Αποτελεί το μεγαλύτερο και πιο ολοκληρωμένο σύνολο δεδομένων οδήγησης μέχρι σήμερα. Όπως φαίνεται και από την ονομασία του συνόλου δεδομένων, αποτελείται από εκατό χιλιάδες εικόνες παρμένες από βίντεο που καταγράφηκαν για να δείχνουν δύσκολες τροχιές οδήγησης σε διάφορες τοποθεσίες στις Ηνωμένες Πολιτείες. Το σύνολο δεδομένων καλύπτει διάφορες καιρικές συνθήκες σε διαφορετικές ώρες της ημέρας, γεγονός που συμβάλλει σημαντικά στην επίτευξη υψηλής ακρίβειας σε ρεαλιστικές συνθήκες.

Το BDD100K μας παρέχει την ευχέρεια να υλοποιήσουμε διάφορες εργασίες μηχανικής όρασης όσων αφορά την οδήγηση όπως εύρεση γραμμών, αναγνώριση αντικειμένων που συναντάμε στο

δρόμο και την εύρεση περιοχών οδήγησης. Ωστόσο, σε αυτή την ενότητα θα ασχοληθούμε μόνο με την εύρεση επιτρεπτής περιοχής οδήγησης εφαρμόζοντας την τεχνική της τμηματικής κατηγοριοποίησης. Για την επίτευξη αυτής της διαδικασίας, το σύνολο δεδομένων BDD100K διαθέτει εβδομήντα χιλιάδες έγχρωμες εικόνες για εκπαίδευση, δέκα χιλιάδες έγχρωμες εικόνες για επαλήθευση μεγέθους 1280 x 720 και τις αντίστοιχες μάσκες που αποτυπώνουν τις περιοχές επιτρεπτής οδήγησης.



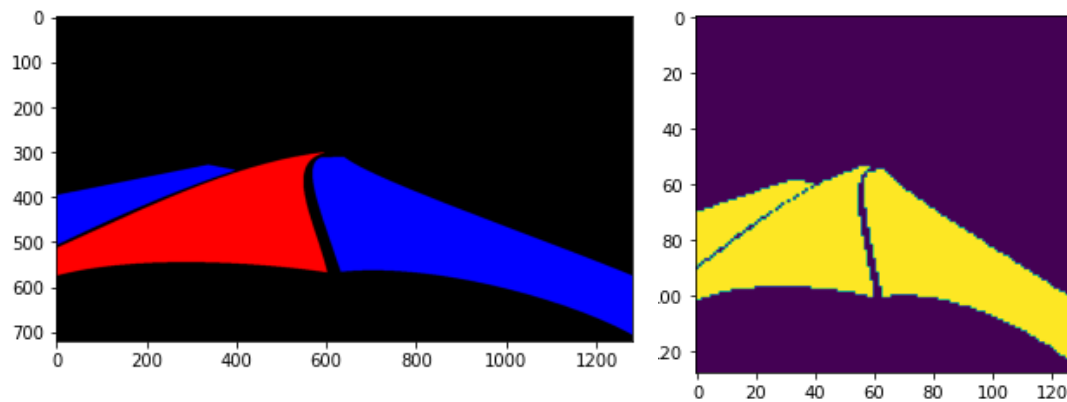
Σχήμα 6.10 Παραδείγματα Εικόνων Επιτρεπτής Περιοχής Οδήγησης Του BDD100K

Στην παραπάνω εικόνα φαίνονται κάποια παραδείγματα από εικόνες στις οποίες έχει εφαρμοστεί η αντίστοιχη μάσκα. Οι περιοχές επιτρεπτής οδήγησης χωρίζονται σε δύο κατηγορίες, στην περιοχή που το όχημα μας έχει προτεραιότητα που απεικονίζεται με κόκκινο χρώμα και στην περιοχή που το όχημα μας μπορεί να κινηθεί αλλά η προτεραιότητα του δρόμου ανήκει δυνητικά σε άλλα οχήματα και απεικονίζεται με μπλε χρώμα.

#### 6.4.2 Προ-Επεξεργασία Δεδομένων

Συνήθως, ένα σύνολο δεδομένων διαθέτει τα δεδομένα του χωρισμένα σε αυτά που προορίζονται για εκπαίδευση, για επαλήθευση και για έλεγχο. Επειδή στη δικιά μας περίπτωση δεν διαθέτουμε δεδομένα για έλεγχο, θα κρατήσουμε για έλεγχο δύο χιλιάδες εικόνες από τις εικόνες που προορίζονται για εκπαίδευση. Με αυτόν τον τρόπο θα μπορούμε να ελέγχουμε την απόδοση του μοντέλου μας σε εικόνες που συναντά πρώτη φορά.

Στη συνέχεια, επειδή τα δεδομένα μας είναι πάρα πολλά και οι υπολογισμοί για την εκπαίδευση του δικτύου είναι χρονοβόρα, αλλάζουμε το μέγεθος της κάθε εικόνας και της αντίστοιχης μάσκας από 1280 x 720 σε 128 x 128. Με αυτόν τον τρόπο, η διαδικασία της εκπαίδευσης γίνεται πολύ πιο γρήγορη και δεν κινδυνεύουμε να εξαντλήσουμε τη διαθέσιμη μνήμη τυχαίας προσπέλασης της GPU.



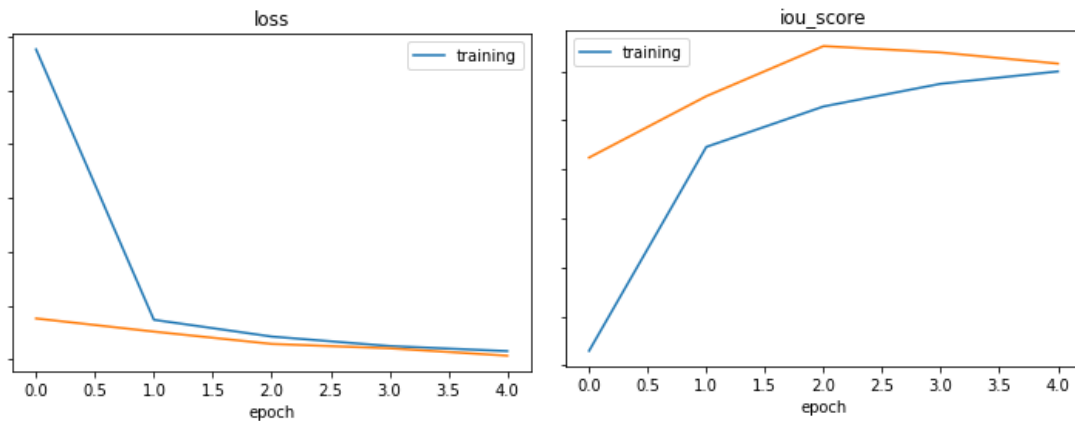
Σχήμα 6.11 Επεξεργασία Των Μασκών Του BDD100K

Τέλος, επειδή εμάς δεν μας ενδιαφέρει τόσο να γνωρίζουμε σε ποιον δρόμο έχει το όχημα μας προτεραιότητα αλλά να εντοπίζουμε απλά ολόκληρη την περιοχή στην οποία μπορεί το όχημά μας να οδηγήσει, επεμβαίνουμε στις μάσκες και τις κάνουμε να έχουν την τιμή μηδέν τα εικονοστοιχεία που δεν επιτρέπεται να οδηγήσει το όχημά μας και ένα αυτά που αντιπροσωπεύουν τις περιοχές που επιτρέπεται.

#### 6.4.3 Εκπαίδευση Δικτύου

Το μοντέλο που επιλέχθηκε για την εκπαίδευση είναι το Unet και δέχεται για είσοδο μια έγχρωμη εικόνα διαστάσεων 128 x 128 και την αντίστοιχη ασπρόμαυρη μάσκα της ίδιων διαστάσεων. Το δίκτυό μας διαθέτει 1.941.139 παραμέτρους προς εκπαίδευση και χρησιμοποιεί τη βαθμολογία IOU για να αξιολογεί τις προβλέψεις του.

Μετά από πολύωρη εκπαίδευση με batch size 16 το μοντέλο μας είναι πλέον σε θέση να πραγματοποιεί τμηματική κατηγοριοποίηση σε εικόνες δρόμου με αρκετά υψηλή ακρίβεια ανεξάρτητα από τις καιρικές συνθήκες που επικρατούν. Η εκπαίδευση διακόπηκε στις 4 εποχές από τη συνάρτηση Early Stopping καθώς η ακρίβεια του μοντέλου σταμάτησε να αυξάνεται.



Σχήμα 6.12 Σχηματική Απεικόνιση Της Πορείας Της Εκπαίδευσης Του Δικτύου

Παραπάνω φαίνονται διαγραμματικά η πορεία της συνάρτησης απώλειας και της ακρίβειας IOU κατά τη διάρκεια της εκπαίδευσης του δικτύου.

#### 6.4.4 Αποτελέσματα Ελέγχου

Πλέον, δίνοντας στο νευρωνικό δίκτυο που εκπαιδεύσαμε μια εικόνα δρόμου μπορεί και μας επιστρέφει την πρόβλεψή του για την αντίστοιχη μάσκα που αντικατοπτρίζει την επιτρεπτή περιοχή οδήγησης. Παρακάτω φαίνεται η εικόνα εισόδου και η πρόβλεψη του δικτύου μας.



Σχήμα 6.13 Πρόβλεψη Επιτρεπτής Περιοχής Οδήγησης

Μένει τώρα να ελέγξουμε κατά πόσο η πρόβλεψη του δικτύου μοιάζει με την μάσκα που προτείνεται από το σύνολο δεδομένων μας.



Σχήμα 6.14 Σύγκριση Εικόνας Πρόβλεψης Με Την Αναμενόμενη Εικόνα Εξόδου

Επειδή τα εικονοστοιχεία της εικόνας που προέβλεψε το δίκτυο και της αναμενόμενης εικόνας έχουν τις τιμές ένα ή μηδέν, ανάλογα με το αν αντιπροσωπεύουν επιτρεπτή περιοχή οδήγησης ή όχι, ο έλεγχος της ακρίβειας του δικτύου θα γίνει με την βοήθεια των μετρικών Precision, Recall και f1-score. Δηλαδή, για κάθε εικονοστοιχείο της αναμενόμενης εικόνας εξόδου θα ελέγχουμε αν το αντίστοιχο της εικόνας πρόβλεψης έχει την ίδια τιμή και στο τέλος ανάλογα με τον αριθμό των σωστών προβλέψεων, οι μετρικές θα μας δείξουν πόσο ακριβές είναι το μοντέλο μας.

Οπότε, το μόνο που έχουμε να κάνουμε είναι να πάρουμε προβλέψεις για καθεμία από τις δύο χιλιάδες εικόνες που κρατήσαμε για τον έλεγχο του μοντέλου, έτσι ώστε ο έλεγχος να γίνει σε εικόνες τις οποίες πρώτη φορά συναντά το δίκτυό μας, να τις συγκρίνουμε με τις αναμενόμενες μάσκες του συνόλου δεδομένων και να πάρουμε το μέσο όρο των τιμών των μετρικών Precision, Recall και f1-score.

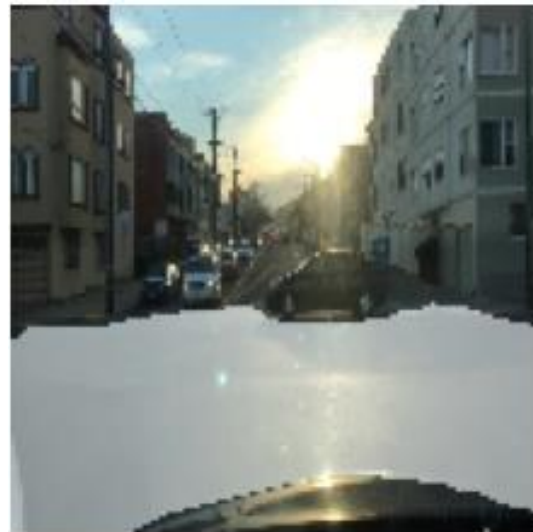
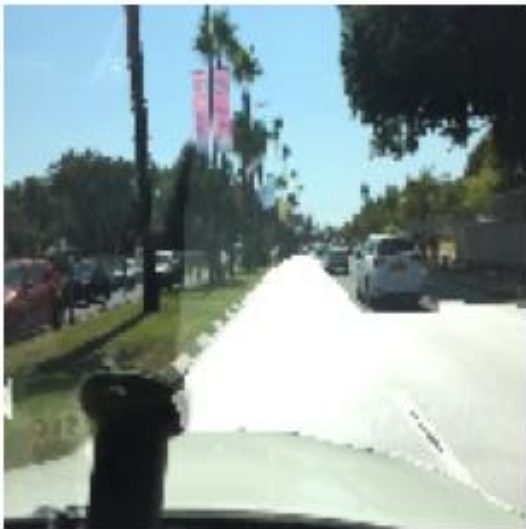
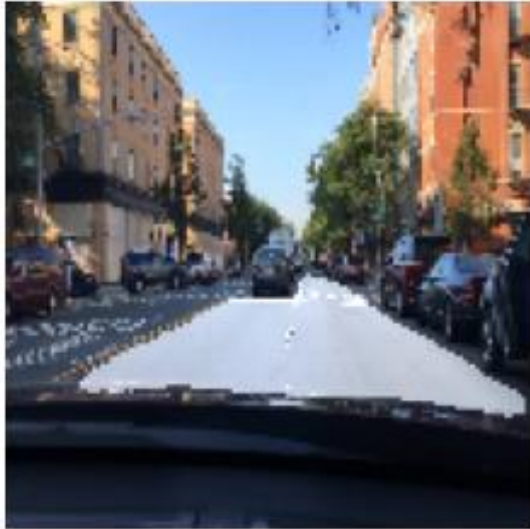
```
Average Precision: 0.8036105211481552
Average Reccall: 0.8430065307435737
Average f1 score: 0.8110077050730975
```

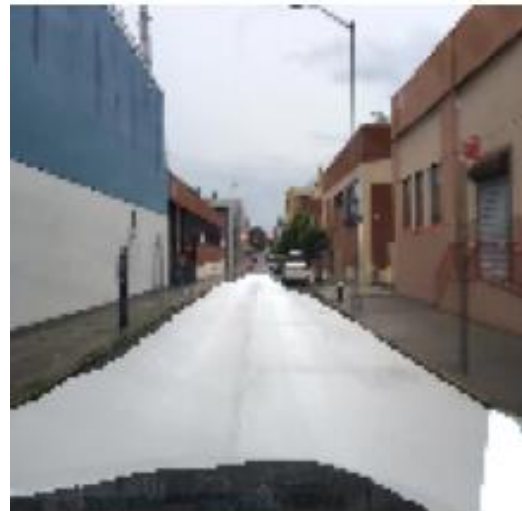
Σχήμα 6.15 Ακρίβεια Προβλέψεων

Όπως φαίνεται, ο μέσος όρος των σωστών προβλέψεων εικονοστοιχείων από τις συνολικές, δίνεται από τη μετρική Precision και είναι 80%, ο μέσος όρος σωστών προβλέψεων εικονοστοιχείων που αντιπροσωπεύουν εικονοστοιχεία επιτρεπτής περιοχής οδήγησης σε σχέση με τα συνολικά που θα έπρεπε να προβλέψει δίνεται από τη μετρική Recall και είναι 84.3%, ενώ ο αρμονικός μέσος όρος των δύο μετρικών που αντικατοπτρίζει και την ακρίβεια του μοντέλου δίνεται από τη μετρική f1-score και είναι 81.1%.



Παρακάτω φαίνονται μερικές προβλέψεις του δικτύου σε δύσκολες εικόνες δρόμου κάτω από διαφορετικές συνθήκες. Για καλύτερη απεικόνιση, απεικονίσαμε την προβλεπόμενη μάσκα στην αρχική εικόνα.





Σχήμα 6.16 Προβλέψεις Δικτύου Σε Δύσκολες Εικόνες Δρόμου



## ΚΕΦΑΛΑΙΟ 7ο

### Σύνοψη και μελλοντικές κατευθύνσεις

#### 7.1 ΣΥΝΟΨΗ

Συνοψίζοντας, σκοπός της παρούσας διπλωματικής εργασίας ήταν η ενασχόληση με τη δημιουργία συστημάτων υποβοήθησης οδήγησης ή ακόμα και αυτόνομης οδήγησης χρησιμοποιώντας αρχιτεκτονικές συνελκτικών νευρωνικών δικτύων. Αναπτύχθηκαν τεχνικές εντοπισμού διαφόρων αντικειμένων που μπορεί να συναντήσει ένα όχημα στο δρόμο καθώς και ο υπολογισμός της απόστασης αυτών από το όχημά μας, βαθιά νευρωνικά δίκτυα ικανά να εντοπίζουν πινακίδες και φωτεινούς σηματοδότες σε εικόνες δρόμου και στη συνέχεια να τις αναγνωρίζουν και τέλος τεχνικές αναγνώρισης της περιοχής που επιτρέπεται το όχημα να οδηγήσει καθώς και παροχή συμβουλών πλοήγησης έτσι ώστε να διατηρηθεί το όχημα στο μέσο της λωρίδας οδήγησης.

Η υλοποίηση έγινε με τη χρήση της γλώσσας προγραμματισμού Python, στη πλατφόρμα Google Colaboratory. Για την ανάπτυξη των νευρωνικών δικτύων χρησιμοποιήθηκαν οι βιβλιοθήκες Keras και Tensorflow, ενώ για την επεξεργασία των εικόνων οι OpenCV και NumPy.

Το πρόβλημα του εντοπισμού αντικειμένων, αποτελεί ένα πολύπλοκο πρόβλημα που καλούνται να λύσουν οι ερευνητές που ασχολούνται με την αυτόνομη οδήγηση καθώς οι προβλέψεις πρέπει να είναι ακριβείς και σε πραγματικό χρόνο. Για τον λόγο αυτό χρησιμοποιήθηκε ένα εκπαιδευμένο σε περισσότερες από 250.000 εικόνες μοντέλο YOLO v5 της Ultralytics που είναι ικανό να εντοπίζει 80 διαφορετικά αντικείμενα που συναντάμε καθημερινά. Ένα από αυτά είναι οι φωτεινοί σηματοδότες. Προκειμένου να αναγνωρίζουμε τι χρώμα είναι ο φωτεινός σηματοδότης, αφού περάσουμε την εικόνα του από τρεις μάσκες, μια για το πράσινο, μια για το κόκκινο και μία για το πορτοκαλί χρώμα, προσπαθούμε να εντοπίσουμε κύκλους σε καθεμία από αυτές. Σε όποια βρεθεί κύκλος, σημαίνει πως αυτό το χρώμα είναι ο φωτεινός σηματοδότης. Στη συνέχεια αναπτύχθηκε μια τεχνική για τον υπολογισμό της απόστασης αυτών των αντικειμένων από την κάμερα η οποία βασίζεται σε μια παραλλαγή του τύπου του εστιακού μήκους. Η μέθοδος αυτή δεν είναι αρκετά ακριβής και αντιμετωπίζει πολλά προβλήματα, ωστόσο μας παρέχει μια γρήγορη εκτίμηση.

Όσον αφορά την αναγνώριση της κυκλοφοριακής σήμανσης, το πρόβλημα είναι σύνθετο και θα πρέπει να διασπαστεί σε δύο επιμέρους, τον εντοπισμό και την ταξινόμηση. Για τον εντοπισμό, έγινε εκπαίδευση ενός δικτύου YOLO v5 για 30 εποχές πάνω σε 852 εικόνες δρόμου που περιέχουν πινακίδες κυκλοφοριακής σήμανσης. Πλέον, αφού είμαστε σε θέση να εντοπίζουμε τις πινακίδες σε μια εικόνα δρόμου, τις τροφοδοτούμε σε ένα δεύτερο νευρωνικό δίκτυο με σκοπό να αναγνωρίσει τι συμβολίζει η καθεμία. Το μοντέλο που επιλέχθηκε έχει 378.023 παραμέτρους και η εκπαίδευση έγινε για 30 εποχές πάνω στο German Traffic Sign Recognition Benchmark σύνολο δεδομένων που περιέχει 51.839 εικόνες πινακίδων χωρισμένες σε 43 διαφορετικές κλάσεις. Έγινε χρήση της τεχνικής data augmentation έτσι ώστε να επιτευχθεί αύξηση και γενίκευση των δεδομένων, με αποτέλεσμα το μοντέλο μας να πετύχει 97,8% ακρίβεια.

Τέλος, ασχοληθήκαμε με την εύρεση της επιτρεπτής περιοχής οδήγησης. Εδώ αναπτύχθηκαν δύο τεχνικές, μία πιο απλή αλλά με όχι τόσο καλά αποτελέσματα και μία αρκετά πιο σύνθετη αλλά πολύ πιο ακριβής. Η πρώτη, ανιχνεύει τις γραμμές στο δρόμο με τη χρήση της τεχνικής HoughLinesP της βιβλιοθήκης του OpenCV και μας παρέχει συμβουλές πλοήγησης με σκοπό να διατηρηθεί το όχημά μας στο μέσο αυτών των γραμμών. Έγινε προσαρμογή σε δρόμους που εντοπίζεται μία γραμμή της λωρίδας οδήγησης, ωστόσο σε δρόμους που δεν υπάρχει διαγράμμιση ή ακόμα και σε στροφές με μεγάλη κλίση, η τεχνική αυτή δε δίνει καλά αποτελέσματα. Η δεύτερη αφορά την εκπαίδευση ενός μοντέλου Unet που κάνει τμηματική κατηγοριοποίηση στην εικόνα. Χρωματίζει δηλαδή όλα τα εικονοστοιχεία που αντιπροσωπεύουν δρόμο στον οποίο επιτρέπεται να οδηγήσει το όχημά μας. Το σύνολο δεδομένων που επιλέχθηκε είναι το Berkley Deep Drive 100K που διαθέτει 80.000 έγχρωμες εικόνες με τις αντίστοιχες μάσκες. Η εκπαίδευση έγινε για 4 εποχές σε 78.000 εικόνες καθώς κρατήσαμε 2.000 για έλεγχο. Ο έλεγχος της ακρίβειας έγινε με τη βοήθεια των μετρικών Precision, Recall και f1-score για τις 2.000 εικόνες και το αποτέλεσμα ήταν 81.1%.

## 7.2 ΜΕΛΛΟΝΤΙΚΕΣ ΚΑΤΕΥΘΥΝΣΕΙΣ

Η ανάπτυξη των αυτόνομων οχημάτων τα τελευταία χρόνια είναι μεγάλη και πλέον είναι το θέμα που απασχολεί την αυτοκινητοβιομηχανία και γενικά την τεχνολογική και ερευνητική κοινότητα. Τα αυτόνομα οχήματα δοκιμάζονται ήδη ενεργά στους δρόμους των Ηνωμένων Πολιτειών Αμερικής. Η υπολογιστική όραση είναι μια από τις κύριες τεχνολογίες που καθιστά δυνατή την ανάπτυξη συστημάτων που χρησιμοποιούνται από τα αυτόνομα οχήματα. Σημαντικό ρόλο σε

αυτό έχει παίξει η αύξηση της υπολογιστικής ισχύς και συγκεκριμένα η σχεδίαση πιο προηγμένων καρτών γραφικών.

Κάποιες μελλοντικές κατευθύνσεις που θα μπορούσαν να εφαρμοστούν μελλοντικά:

- Δημιουργία μεγαλύτερων συνόλων δεδομένων όσον αφορά τον εντοπισμό και την αναγνώριση της κυκλοφοριακής σήμανσης έτσι ώστε να γίνει η εκπαίδευση ενός μόνο δικτύου. Κάτι που θα συμβάλλει στην επίτευξη πιο γρήγορων προβλέψεων.
- Πρόβλεψη κυκλοφοριακής κίνησης με Graph Neural Networks.
- Εύρεση οικονομικότερων λύσεων σε ότι αφορά τη δημιουργία συστημάτων αυτόνομης οδήγησης έτσι ώστε να μπορεί οποιοσδήποτε να αγοράσει ένα τέτοιο όχημα.
- Χρήση άλλων αρχιτεκτονικών νευρωνικών δικτύων για επίτευξη καλύτερων αποτελεσμάτων.
- Ανάπτυξη μιας πιο ακριβούς μεθόδου υπολογισμού των αποστάσεων.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

---

- [1] J. McCarthy, σε *Automata Studies*, 1956, pp. 177-182.
- [2] I. C. L. Hub, «What is Supervised Learning?», 19 Αύγουστος 2020. [Ηλεκτρονικό]. Available: <https://www.ibm.com/cloud/learn/supervised-learning>.
- [3] Mathworks, «What Is Machine Learning?», 2019. [Ηλεκτρονικό]. Available: <https://uk.mathworks.com/discovery/machine-learning.html#machine-learning-with-matlab>.
- [4] M. W. v. Otterlo, Wiering et al. *Reinforcement Learning*, 2012.
- [5] F. Chollet, *Deep Learning with Python*, New York: Manning, 2018.
- [6] M. Sazli, A brief review of feed-forward neural networks, University of Ankara, 2006, pp. 11-17.
- [7] F. Rosenblatt, The perceptron: A probabilistic model for information storage and, τόμ. 65, 1958, pp. 386-408.
- [8] S. Haykin, *Neural Networks*, New York, NY: , 1998.
- [9] I. S. κ. O. V. W. Zaremba, «Recurrent Neural Network Regularization», Φεβρουάριος 2015.
- [10] S. SHARMA, «Towards Data Science», 6 Σεπτέμβριος 2017. [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>.
- [11] R. R. κ. R. J. Marks, *Classical Optimization Techniques*, Cambridge, 1999.
- [12] I. Changhau, *Loss Functions in Neural Networks*, 2017.
- [13] S. Ruder, «An overview of gradient descent optimization algorithms, 2017.
- [14] K. Fukushima, «Neocognitron: A self-organizing neural network model for a», τόμ. 36, 1980, pp. 193-202.
- [15] M. Basavarajaiah, «medium», 30 Μάρτιος 2019. [Ηλεκτρονικό]. Available: <https://medium.com/@bdhuma/6-basic-things-to-know-about-convolution-daef5e1bc411>.
- [16] J. Brownlee, «Machine Learning Mastery», 5 Ιούλιος 2019. [Ηλεκτρονικό]. Available: <https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>.
- [17] S. A. A. S. S. K. κ. H. R. M. Bennamoun, *A Guide to Convolutional Neural Networks for Computer Vision*, Morgan & Claypool, 2018.
- [18] C. C. Aggarwal, *Neural networks and deep learning: a textbook*, 2018, pp. 327-328.

- [19] S. Team, «SuperDataScience,» 18 Αύγουστος 2018. [Ηλεκτρονικό]. Available: <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening>.
- [20] D. M. Hawkins, The Problem of Overfitting, 2003.
- [21] I. S. a. G. E. H. A. Krizhevsky, Advances in Neural, 2012.
- [22] [Ηλεκτρονικό]. Available: <https://colab.research.google.com/>.
- [23] L. Hulstaert, «A Beginner's Guide to Object Detection,» 19 Απρίλιος 2018. [Ηλεκτρονικό]. Available: <https://www.datacamp.com/community/tutorials/object-detection-guide>.
- [24] M. Chablani, «Towards Data Science,» 21 Αύγουστος 2017. [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006>.
- [25] ultralytics. [Ηλεκτρονικό]. Available: <https://github.com/ultralytics/yolov5>.
- [26] «kaggle,» [Ηλεκτρονικό]. Available: <https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign>.
- [27] J. Zhang, «Towards Data Science,» 18 Οκτώβριος 2019. [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/unet-line-by-line-explanation-9b191c76baf5>.
- [28] SEONGJUHONG, 08 12 2019. [Ηλεκτρονικό].
- [29] K. Markham, «What is machine learning, and how does it work?,» 2018. [Ηλεκτρονικό].
- [30] T. G. Blog, «What is the Difference between Classification and Regression,» 1 Ιανουάριος 2021. [Ηλεκτρονικό]. Available: <https://kindsonthegenius.com/blog/what-is-the-difference-between-classification-and-regression>.
- [31] A. J. Smola, «Alexander J. Smola,» 2008. [Ηλεκτρονικό]. Available: <http://www.hutter1.net/mlss08/files/smola.pdf>.
- [32] O. I. T. O. Ugurlu U, Ugurlu U, Oksuz I, Tas O, 2018.
- [33] V. Mishra, «ResearchGate,» Ιανουάριος 2018. [Ηλεκτρονικό]. Available: [https://www.researchgate.net/figure/Recurrent-neural-networkRNN-or-Long-Short-Term-MemoryLSTM-5616\\_fig2\\_324883736](https://www.researchgate.net/figure/Recurrent-neural-networkRNN-or-Long-Short-Term-MemoryLSTM-5616_fig2_324883736).
- [34] abhigoku10, «Topic DL01: Activation functions and its Types in Artificial Neural network,» 27 Μάρτιος 2018. [Ηλεκτρονικό]. Available: <https://abhigoku10.medium.com/activation-functions-and-its-types-in-artificial-neural-network-14511f3080a8>.
- [35] A. Suman, «medium,» 26 Αύγουστος 2020. [Ηλεκτρονικό]. Available: <https://medium.com/analytics-vidhya/activation-function-c762b22fd4da>.
- [36] R. Phung, «A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets,» Οκτώβριος 2019.

- [37] A. Ali, «medium,» 22 Μάιος 2019. [Ηλεκτρονικό]. Available: <https://medium.com/machine-learning-researcher/convlutional-neural-network-cnn-2fc4faa7bb63>.
- [38] «computersciencewiki,» [Ηλεκτρονικό]. Available: [https://computersciencewiki.org/index.php/Max-pooling/\\_/\\_Pooling](https://computersciencewiki.org/index.php/Max-pooling/_/_Pooling).
- [39] «The Dangers of Under-fitting and Over-fitting,» 3 Δεκέμβριος 2019. [Ηλεκτρονικό]. Available: <https://medium.com/analytics-vidhya/the-dangers-of-under-fitting-and-over-fitting-495f9efa1847>.
- [40] «WIKIMEDIA COMMONS,» [Ηλεκτρονικό]. Available: [https://commons.wikimedia.org/wiki/File:Intersection\\_over\\_Union\\_-\\_poor,\\_good\\_and\\_excellent\\_score.png](https://commons.wikimedia.org/wiki/File:Intersection_over_Union_-_poor,_good_and_excellent_score.png).
- [41] «lotus-qa,» [Ηλεκτρονικό]. Available: <https://www.lotus-qa.com/services/image-annotation-semantic-segmentation/>.