



ΔΙΕΘΝΕΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΕΛΛΑΔΟΣ

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ  
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ,  
ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

**ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΣΩΠΩΝ ΣΕ ΒΙΝΤΕΟ  
ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ**

**Πτυχιακή Εργασία της**  
Ελένης Καραγιάννη (3498)

Επιβλέπων: Στ. Βολογιαννίδης, Επίκουρος Καθηγητής

**ΣΕΡΡΕΣ, ΙΑΝΟΥΑΡΙΟΣ 2020**

**Υπεύθυνη Δήλωση**: Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης, βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών του Διεθνούς Πανεπιστημίου της Ελλάδας.



## Περιεχόμενα

Εισαγωγή .....	6
1. Βάση Μηχανική Μάθησης και Open-Source Πλατφόρμες Αναγνώρισης Προσώπου .....	8
1.1. Τεχνητή νοημοσύνη και μηχανική μάθηση .....	8
1.2. Κύριοι Τύποι Μηχανικής Μάθησης .....	10
1.3. Νευρωνικά Δίκτυα .....	13
1.3.1. Βιολογικά Νευρωνικά Δίκτυα .....	13
1.3.2 Τεχνητά Νευρωνικά Δίκτυα .....	14
1.3.2.1 Δομή Νευρώνα.....	14
1.3.2.2. Δομή Τεχνητών Νευρωνικών Δικτύων.....	17
1.3.2.3. Αρχιτεκτονική Τεχνητών Νευρωνικών Δικτύων.....	17
1.3.2.4. Πλεονεκτήματα και Μειονεκτήματα Τεχνητών Νευρωνικών Δικτύων .....	18
1.4. Βαθιά Μάθηση (Deep Learning) – Τι είναι και πως λειτουργεί στα ΤΝΔ;.....	20
1.4.1 Τι είναι η Βαθιά Μάθηση .....	20
1.4.2. Deep Learning vs Machine Learning.....	20
1.4.3. Πως Λειτουργεί.....	21
1.4.4. Εφαρμογές της Βαθιάς Μάθησης .....	23
1.4.5. Απαιτήσεις Hardware .....	23
1.5. Συνελκτικά Νευρωνικά Δίκτυα (Convolutional Neural Network - CNNs).....	25
1.6. Αναγνώριση Προσώπου.....	30
1.6.1. Πως λειτουργεί.....	30
1.6.2. Ιστορική Αναδρομή .....	31
1.6.3 Αποτελέσματα Ερευνών .....	32
1.6.4. Εφαρμογές στην καθημερινότητα.....	33
1.6.5. Πλεονεκτήματα και Μειονεκτήματα .....	34

1.7. Open-Source Πλατφόρμες Αναγνώρισης Προσώπου.....	36
1. DeepFace.....	37
2. CompreFace .....	38
3. Face Recognition .....	39
4. InsightFace.....	40
5. FaceNet .....	41
6. InsightFace-REST.....	42
Αναφορά σε άλλες εφαρμογές.....	43
2. Η πλατφόρμα OpenCV σε γλώσσα προγραμματισμού Python .....	44
2.1. OpenCV .....	44
2.2. Χαρακτηριστικά τύπου Haar .....	48
2.3. Εικόνες Ακεραίων Αριθμών (Integral Images).....	50
2.4. Αλγόριθμος AdaBoost .....	52
2.5. Καταρράκτης Ταξινομητών.....	53
3. Παρουσίαση Project.....	54
Συμπεράσματα .....	84
Βιβλιογραφία .....	85

## Εισαγωγή

Η Αναγνώριση Προσώπου (facial recognition) τόσο σε βίντεο όσο και σε εικόνα αποτελεί μία από τις πλέον διαδεδομένες Βιομετρικές Τεχνολογίες, η οποία παρέχει τη δυνατότητα της αναγνώρισης ενός ατόμου μέσω της ψηφιακής του απεικόνισης (φωτογραφία ή βίντεο). Άλλες Βιομετρικές Τεχνολογίες είναι η σάρωση της ίριδας του ματιού καθώς και η σάρωση των δακτυλικών αποτυπωμάτων.

Τα συστήματα Αναγνώρισης Προσώπου έχουν αποκτήσει μεγάλη δημοτικότητα τα τελευταία χρόνια, καθώς παρέχουν μια ακριβή μέθοδο για τον εντοπισμό ατόμων και για λόγους ασφαλείας (καταπολέμηση εγκληματικότητας, διασφάλιση δημόσιας τάξης κλπ.).

Ραγδαία εξέλιξη παρατηρείται στην τεχνολογία αυτή τα τελευταία χρόνια, η οποία οφείλεται στο γεγονός ότι τόσο οι μέθοδοι αναγνώρισης όσο και οι αλγόριθμοι που εφαρμόζονται για την επεξεργασία των δεδομένων υλοποιούνται πιο εύκολα. Αυτό συμβαίνει γιατί πλέον έχουμε εύκολα πρόσβαση σε πληθώρα δεδομένων (π.χ. εικόνες/φωτογραφίες).

Συγκεκριμένα, ύστερα από τα γεγονότα της 11ης Σεπτεμβρίου που έλαβαν μέρος στις Ηνωμένες Πολιτείες της Αμερικής και με τις επαναλαμβανόμενες απειλές για την ασφάλεια, η Αμερικανική Κυβέρνηση κάνοντας μια τεράστια χρηματοδότηση βοήθησε στην ραγδαία ανάπτυξη της τεχνολογίας αυτής.

Το 1964 αποτελεί ορόσημο για την τεχνολογία Αναγνώρισης Προσώπου, με πρωτοπόρο τον Αμερικάνο μαθηματικό και επιστήμονα υπολογιστών Woody Bledsoe. Ο Woody κάνοντας χρήση ενός software και μίας βάσης δεδομένων φωτογραφιών, κατάφερε να ταυτοποιήσει επιτυχώς την φωτογραφία ενός υπόπτου με μία από τις ήδη υπάρχουσες στη βάση δεδομένων.

Στο πρώτο κεφάλαιο γίνεται αναφορά σε κάποιες βασικές έννοιες και ορισμούς, οι οποίες επικεντρώνονται στην μηχανική μάθηση. Γίνεται επεξήγηση των εννοιών «Τεχνητή Νοημοσύνη», «κατηγορίες της Μηχανικής Μάθησης», «Νευρωνικά Δίκτυα» και πως λειτουργούν. Επίσης παρουσιάζεται η συσχέτιση που έχουν και γίνεται ανάλυση για τα «Συνελκτικά Δίκτυα». Μετά την επεξήγηση της σημασίας που έχουν, διατυπώνεται ο τρόπος αναγνώρισης ενός προσώπου σε μια εικόνα ή ένα βίντεο σε πραγματικό χρόνο. Στην συνέχεια αναλύονται: η «Βιομετρική Ταυτοποίηση», οι τεχνικές και τα βήματα που ακολουθούνται. Στο τέλος του κεφαλαίου αναφέρονται ορισμένα συμπεράσματα και δίνονται μερικά παραδείγματα από πλατφόρμες αναγνώρισης προσώπου ανοιχτού κώδικα. Στο δεύτερο κεφάλαιο της εργασίας γίνεται ανάλυση

της πλατφόρμας OpenCV μέσω Python και των χαρακτηριστικών Αναγνώρισης Προσώπου μέσω Βίντεο, επίσης αναφέρονται τα χαρακτηριστικά Haar και ο αλγόριθμος AdaBoost.

## 1. Βάση Μηχανική Μάθησης και Open-Source Πλατφόρμες Αναγνώρισης Προσώπου

### 1.1. Τεχνητή νοημοσύνη και μηχανική μάθηση

Το 1950 ο Άγγλος μαθηματικός Alan Turing δημοσίευσε το άρθρο «Υπολογιστικά Μηχανήματα και Νοημοσύνη»[1], [2] στο οποίο γίνεται αναφορά για το Παιχνίδι της Μίμησης (πλέον γνωστό ως δοκιμασία Turing)[3]. Η επεξήγηση του «Παιχνιδιού της Μίμησης» είναι η εξής: Μέσα σε ένα δωμάτιο βρίσκεται ένας άντρας (Α) και μια γυναίκα (Β) οι οποίοι επικοινωνούν με μέσω τηλετύπου με έναν άνθρωπο (Γ) ο οποίος βρίσκεται σε ένα άλλο δωμάτιο. Στόχος του παιχνιδιού είναι μέσα από την επικοινωνία που έχει ο (Γ) με τους (Α) και (Β) να μαντέψει ποιος από τους δύο είναι γυναίκα. Να σημειωθεί ότι ο (Α) προσπαθεί να παραπλανήσει τον (Γ) ότι είναι γυναίκα, ενώ η (Β) προσπαθεί να τον βοηθήσει και να τον πείσει ότι όντως είναι γυναίκα. Δηλαδή και οι 2 παίκτες (Α) και (Β) προσπαθούν να πείσουν τον (Γ) ότι είναι γυναίκες και ο (Γ) προσπαθεί να καταλάβει ποιος λέει την αλήθεια.

Το ερώτημα του Turing είναι : τι θα συμβεί αν στην θέση του (Α) βάλουμε μια μηχανή ; Ο (Γ) θα έχει το ίδιο ποσοστό αποτυχίας στην περίπτωση αυτή σε σύγκριση με την περίπτωση όπου (Α) και (Β) ήταν άνθρωποι ; Με τον τρόπο αυτό πρωτοεμφανίστηκε η Τεχνητή Νοημοσύνη.

Ο όρος «Τεχνητή Νοημοσύνη» διατυπώνεται για πρώτη φορά το 1956 στο Dartmouth College κατά τη διάρκεια ενός ερευνητικού συνεδρίου από τον John McCarthy ως «η επιστήμη και η μηχανική κατασκευής ευφύων μηχανών».

Η τεχνητή νοημοσύνη ασχολείται με την εφαρμογή των προγραμμάτων τα οποία μπορούν εμφανίζουν στοιχεία της ανθρώπινης συμπεριφοράς και να τα μιμούνται, όπως είναι: η αναγνώριση, η μνήμη, η μάθηση, η επίλυση προβλημάτων, η κατανόηση της φυσικής γλώσσας, προσαρμοστικότητα, αναγνώριση αντικειμένων κ.α.. Με παρόμοιο τρόπο και οι μηχανές μπορούν να αναπτύξουν ένα παραπλήσιο τρόπο αντίληψης και αλληλεπίδρασης με το περιβάλλον μέσω της αναγνώρισης προτύπων, όπως είναι η επίλυση σύνθετων προβλημάτων, η κατανόηση του χώρου (π.χ. χρησιμοποιώντας κάμερες) ή η επίτευξη ενός στόχου που τους έχει τεθεί.

Ο προγραμματισμός ενός υπολογιστή για την κατανόηση του περιβάλλοντός του, απαιτεί τη συμμετοχή αρκετών επιστημονικών αντικειμένων, όπως είναι : η επεξεργασία της φυσικής γλώσσας, η ρομποτική, η μηχανική όραση, και άλλες επιστήμες οι οποίες τοποθετούνται στο ευρύ πλαίσιο της σύγχρονης νοημοσύνης ως ανεξάρτητα πεδία της.

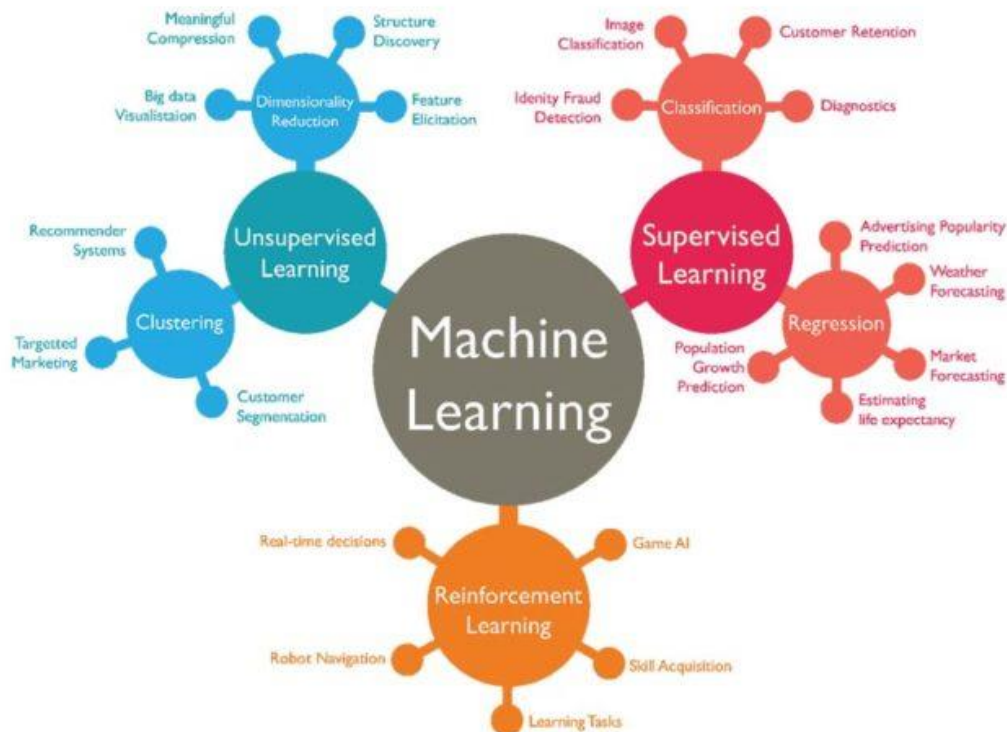


Το πεδίο της Τεχνητής Νοημοσύνης[4] θεωρείται μια από τις νεότερες επιστήμες. Σε αυτό υπάρχει το υπό-πεδίο της Μηχανικής Μάθησης που βασίζεται στη δυνατότητα των συστημάτων να μαθαίνουν, να βελτιώνουν δηλαδή την απόδοσή τους σε κάποιους τομείς μέσω της αξιοποίησης προηγούμενης γνώσης και εμπειρίας που έχουν αποκτήσει. Όπως επίσης να έχουν την ικανότητα να παίρνουν αποφάσεις με την ελάχιστη ανθρώπινη παρέμβαση (σύμφωνα με αλγορίθμους). Το 1959, ο Άρθουρ Σάμουελ ορίζει τη Μηχανική Μάθηση ως "Πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί". Ένας σχετικός γενικός ορισμός Μηχανικής Μάθησης[5] δίνεται από τον Mitchell (1997): "Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από την εμπειρία  $E$  σε σχέση με κάποια εργασία  $T$  και κάποιο μέτρο απόδοσης  $P$ , εάν η απόδοσή του στο  $T$ , όπως μετριέται από το  $P$ , βελτιώνεται με την εμπειρία  $E$ "

Στην καθημερινότητα μπορεί να γίνει αναφορά σε μερικούς τομείς όπου εφαρμόζεται η Μηχανική Μάθηση, όπως είναι : το Marketing, η χρηματιστηριακή ανάλυση, η ανάκτηση πληροφοριών, η αναγνώριση εικόνων / αντικειμένων, η ιατρική διάγνωση διάφορων παθήσεων, ο εντοπισμός κακόβουλου λογισμικού, η Βιοπληροφορική και άλλα.

## 1.2. Κύριοι Τύποι Μηχανικής Μάθησης

Ο τομέας της Μηχανικής Μάθησης[5]–[7] αναπτύσσεται σε τρεις κύριες κατηγορίες μάθησης, όπως είναι και αυτές που επιτρέπουν στον άνθρωπο να αποκτήσει γνώσεις με έναν παρόμοιο τρόπο. Αυτές είναι: η Επιβλεπόμενη Μάθηση (Supervised Learning), η Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning) και η Ενισχυτική Μάθηση (Reinforcement Learning). Υπάρχουν βέβαια και υποκατηγορίες, όπως είναι η Ημι-επιβλεπόμενη Μάθηση (Semi-supervised Learning) και η Εξελικτική Μάθηση (Evolutionary Learning).



1. Βασικές κατηγορίες Μηχανικής Μάθησης (Επιβλεπόμενη Μάθηση, Μη Επιβλεπόμενη Μάθηση, Ενισχυτική Μάθηση)

**Επιβλεπόμενη Μάθηση (Supervised Learning):** Είναι η διαδικασία μάθησης που δίνει το αποτέλεσμα σε μια συνάρτηση ενός υπολογιστικού προγράμματος. Έτσι από το σύνολο των παραδειγμάτων σωστής συμπεριφοράς αρχίζει ο αλγόριθμος να τα σχετίζει με τις επιθυμητές εξόδους. Ο αλγόριθμος από έναν εξωτερικό επιβλέποντα (external supervisor) μαθαίνει τις

επιθυμητές συμπεριφορές, ώστε στις επόμενες εισόδους να ξέρει τι θα πρέπει να ακολουθήσει, δηλαδή να προβλέψει τις σωστές εξόδους. Ένα μειονέκτημα που παρατηρείται σε αυτού του είδους μάθηση είναι ότι χρειάζεται πολυάριθμα παραδείγματα ώστε να δημιουργηθεί ένα ικανοποιητικό προβλεπτικό μοντέλο.

Επέκταση αυτής της κατηγορίας Μηχανικής Μάθησης αποτελεί η Ημι-επιβλεπόμενη Μάθηση (Semi-supervised Learning), η οποία χρησιμοποιεί ένα μικρό αριθμό δεδομένων με επιθυμητή έξοδο και ένα μεγάλο αριθμό χωρίς να γνωρίζουμε αν θα υπάρξει επιθυμητή έξοδος.

Η Επιβλεπόμενη Μάθηση χρησιμοποιείται για την επίλυση προβλημάτων, όπως είναι τα ακόλουθα:

1. Ταξινόμησης (Classification): για τη δημιουργία μοντέλων πρόβλεψης διακριτών τάξεων, όπως π.χ. η ομάδα αίματος.
2. Παλινδρόμησης (Regression): για τη δημιουργία μοντέλων πρόβλεψης αριθμητικών τιμών, με τη πρόβλεψη της τιμής όσο πιο κοντά γίνεται στην πραγματική της έξοδο.

Ένα παράδειγμα χρήσης της Επιβλεπόμενης Μάθησης που συναντάμε με μεγάλο ποσοστό είναι οι προβλέψεις της αγοράς, δηλαδή την ζήτηση των προϊόντων, ποια θα έχουν περισσότερη προτίμηση από τους πελάτες. Όπως και οι προβλέψεις των τιμών σε αρκετά προϊόντα.

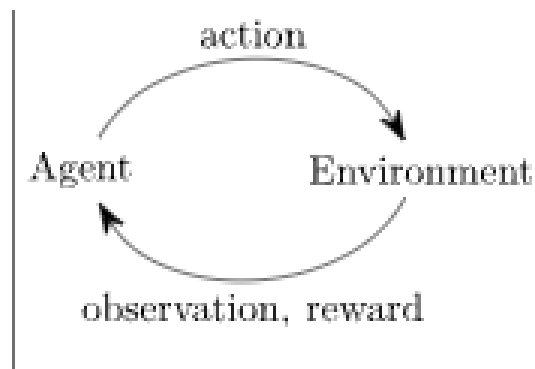
**Μη Επιβλεπόμενη Μάθηση (Unsupervised learning):** Είναι η κατηγορία μάθησης που επιτρέπει το σύνολο των εισόδων (δεδομένα) χωρίς να είναι προσδιορισμένα, συνεπώς δε είναι δυνατή η καθοδήγηση, όπως στην προηγούμενη κατηγορία μάθησης. Έτσι είναι αναμενόμενη η εμπειρία των αλγορίθμων ή κάποιο συγκεκριμένο πρότυπο να μην είναι υπαρκτό. Στα επιθυμητά αποτελέσματα δεν δίνονται συγκεκριμένες τιμές εξόδων, υπάρχει όμως η ομαδοποίηση σύμφωνα με μοτίβα, ομοιότητες και διαφορές, χωρίς κάποια προηγούμενη εκπαίδευση. Συνεπώς γίνεται μάθηση μέσω παρατήρησης.

Η Μη Επιβλεπόμενη Μάθηση έχει σαν τρόπους αντιμετώπισης των προβλημάτων τα εξής:

1. Ανάλυσης Συσχετισμών (Association Analysis)
2. Ομαδοποίησης (Clustering): Σκοπός είναι ο καλύτερος διαχωρισμός σε ομάδες. Αφού γίνει αρχικά μια διερεύνηση, γίνεται η ομαδοποίηση ανάλογα με τις ομοιότητες και τις διαφορές στα χαρακτηριστικά των δεδομένων. Η διαφορά με την ταξινόμηση είναι ότι δεν είναι γνωστό πόσες και ποιες ομάδες αλγορίθμων υπάρχουν.
3. Μείωση διάστασης (Dimensionality reduction)

Παραδείγματα χρήσης της Μη Επιβλεπομένης μάθησης αποτελούν οι διαφημίσεις. Ανάλογα με τις προτιμήσεις των χρηστών στο διαδίκτυο γίνεται παρατήρηση και λήψη πληροφοριών για την καλύτερη πλοήγηση των χρηστών σύμφωνα με τα δεδομένα τους, καθώς και καλύτερη παροχή υπηρεσιών (διότι έτσι γίνεται ανατροφοδότηση πελάτη - εταιρίας/υπηρεσίας). Ο τομέας αυτός εντάσσεται στο ψηφιακό Marketing. Ένα αξιοσημείωτο παράδειγμα είναι αυτό των προτεινόμενων ταινιών ή σειρών σύμφωνα με τις δίκες μας προτιμήσεις προβολών.

**Ενισχυτική Μάθηση (Reinforcement Learning):** Αποτελεί μια τεχνική Μηχανικής Μάθησης όπου το σύστημα μάθησης προσπαθεί να μάθει μέσω της άμεσης αλληλεπίδρασης με το περιβάλλον. Στόχος του συστήματος είναι η μεγιστοποίηση της ανταμοιβής (reward) που λαμβάνει. Η οντότητα η οποία μαθαίνει αποκαλείται «πράκτορας» (agent), οτιδήποτε άλλο διαφορετικό από αυτό είναι μέρος του περιβάλλοντος. Το περιβάλλον και ο πράκτορας βρίσκονται σε συνεχή αλληλεπίδραση μεταξύ τους, με το πρώτο να επιστρέφει ανταμοιβές (rewards) στο δεύτερο και ο δεύτερος να επιλέγει ενέργειες (actions). Ο πράκτορας μαθαίνει από τις προηγούμενες εμπειρίες του με αποτέλεσμα σε βάθος χρόνου να επιλέγει βέλτιστα τις ενέργειες (actions). Αυτό συμβαίνει με την βοήθεια των ανταμοιβών (rewards) που λαμβάνονται στην κατάσταση που βρίσκεται ο πράκτορας.



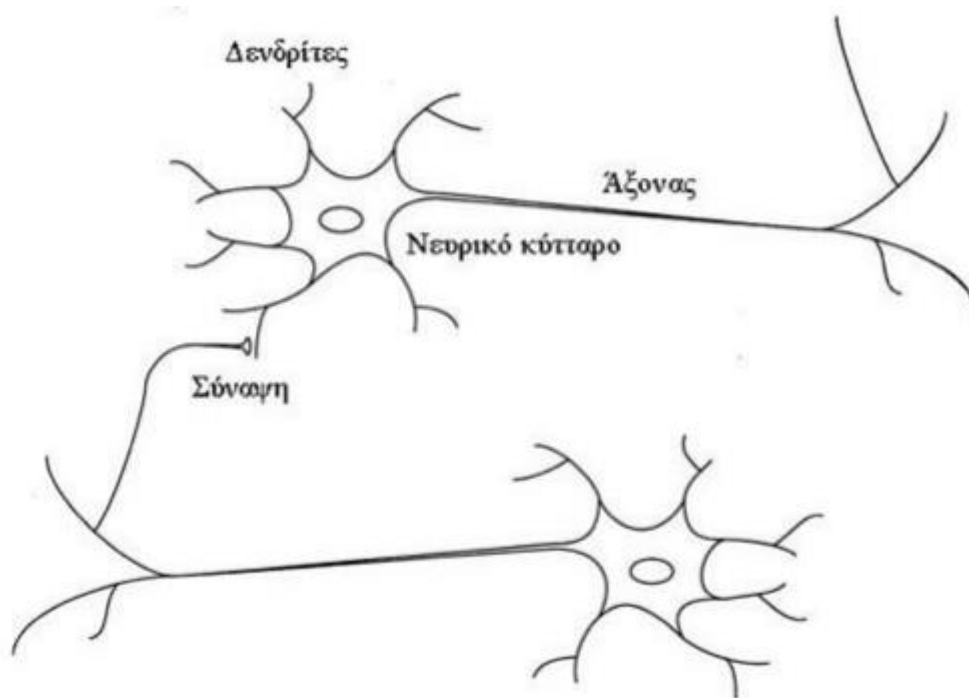
2. Λειτουργία Ενισχυτικής Μάθησης

Χρησιμοποιείται κυρίως για την επίλυση προβλημάτων Σχεδιασμού (Planning), όπως π.χ. ο έλεγχος κίνησης ρομπότ, η βελτιστοποίηση ενεργειών/εργασιών σε εργοστάσια κλπ.. Επίσης, η τεχνική αυτή είναι πολύ σημαντική για τα σημερινά παιχνίδια.

### 1.3. Νευρωνικά Δίκτυα

#### 1.3.1. Βιολογικά Νευρωνικά Δίκτυα

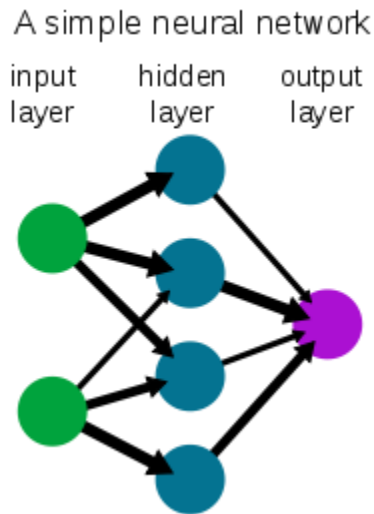
Νευρωνικό Δίκτυο (ΝΔ) αποκαλείται ένα κύκλωμα από διασυνδεδεμένους νευρώνες. Τον 19ο αιώνα έγινε γνωστό από τους επιστήμονες ότι η δομή του εγκεφάλου απαρτίζεται από διακριτά δομικά στοιχεία (ή μονάδες), τους νευρώνες. Κάθε νευρώνας αποτελείται από: το σώμα, τους δενδρίτες και τον άξονα. Μέσα στο σώμα του κάθε νευρώνα γίνονται όλες οι απαραίτητες διαδικασίες/ υπολογισμοί. Οι δενδρίτες είναι υπεύθυνοι για την επικοινωνία του κάθε νευρώνα με τους υπόλοιπους, συλλέγοντας/ λαμβάνοντας ηλεκτρικά σήματα (νευρικές ώσεις). Ο άξονας είναι η έξοδος του κάθε νευρώνα, αποτελεί δηλαδή την απόφαση του εγκεφάλου αφού υλοποιηθεί η επεξεργασία των ερεθισμάτων. Η δομή των αξόνων ενός κυττάρου διασφαλίζει την διάδοση/ αποστολή ηλεκτρικών παλμών κατά μήκος του άξονα με ταχύτητα 100m/s. Η σύνδεση των νευρώνων γίνεται μέσω των συνάψεων. Με απλά λόγια ένα Νευρωνικό Δίκτυο είναι ο τρόπος λειτουργίας του ανθρώπινου εγκεφάλου.



3. Βιολογικά Νευρωνικά Δίκτυα - Δομή Βιολογικού Νευρώνα

### 1.3.2 Τεχνητά Νευρωνικά Δίκτυα

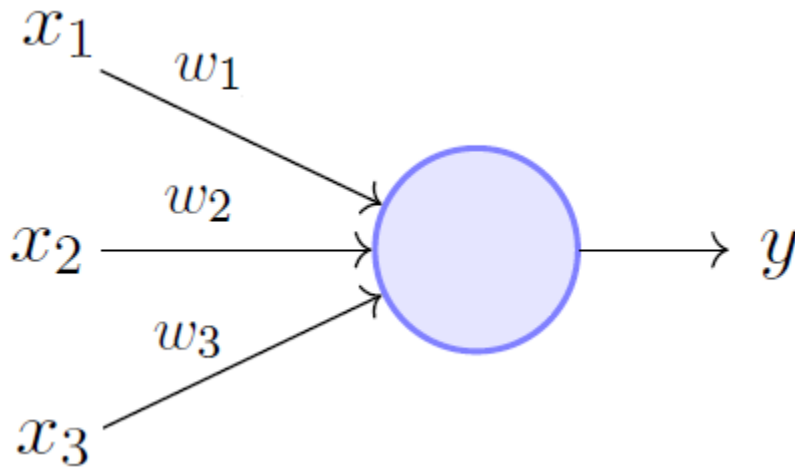
Τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ)[8] προσομοιάζουν την λειτουργία του ανθρώπινου εγκεφάλου. Στην ουσία πρόκειται για ένα σύστημα επεξεργασίας δεδομένων, το οποίο αποτελείται από διακριτές δομικές μονάδες τους τεχνητούς νευρώνες. Απαρτίζεται από τρία βασικά επίπεδα: το επίπεδο εισόδου όπου εισάγονται τα δεδομένα, το επίπεδο εξόδου όπου εξάγεται το αποτέλεσμα και ανάμεσά τους τα κρυμμένα επίπεδα όπου γίνεται η επεξεργασία των δεδομένων. Βασικό χαρακτηριστικό των ΤΝΔ είναι η δυνατότητα εκπαίδευσης τους μέσα από μια διαδικασία μηχανικής μάθησης. Αυτό τα καθιστά ιδιαίτερα ισχυρά εργαλεία για την επίλυση αρκετά δύσκολων και σύνθετων προβλημάτων.



4. Επίπεδα Τεχνητού Νευρωνικού Δικτύου

#### 1.3.2.1 Δομή Νευρώνα

Το πιο απλό παράδειγμα υπολογιστικού νευρώνα είναι το perceptron, πρόκειται για ένα είδος ΤΝΔ το οποίο εφευρέθηκε το 1957 από τον Frank Rosenblatt. Έχει έναν αριθμό εισόδων (αντίστοιχα δενδρίτες) και κάθε μια από τις εισόδους έχει μια τιμή βάρους. Μέσα στο σώμα του νευρώνα γίνεται ο υπολογισμός μιας συνάρτησης και η έξοδος της συνάρτησης αποτελεί την έξοδο του νευρώνα.



5. Παράδειγμα υπολογιστικού νευρώνα - Perceptron

Στο παραπάνω σχήμα: Ο κύκλος αντιπροσωπεύει τον νευρώνα, ο οποίος έχει πολλές εισόδους ( $X_1, X_2, X_3 \dots X_n$ ) με τα αντίστοιχα βάρη ( $W_1, W_2, W_3 \dots W_n$ ) και μία έξοδο ( $Y$ ).

Συγκεκριμένα, όπως βλέπουμε και στο σχήμα κάθε είσοδος/σήμα ( $X_1, X_2, X_3 \dots X_n$ ) χαρακτηρίζεται από μία μοναδική τιμή βάρους ( $W_1, W_2, W_3 \dots W_n$ ) η οποία καθορίζει την σημαντικότητα της κάθε εισόδου/σήματος. Συμπερασματικά, είσοδος ενός νευρώνα είναι ο συνδυασμός των εισόδων/σημάτων σύμφωνα με τον τύπο και την λειτουργία που πρέπει να εκτελέσει το κάθε ένα. Η έξοδος ( $Y$ ) είναι μοναδική και αποτελεί συνάρτηση των εισόδων/σημάτων εισόδου.

Με το που ενεργοποιείται ο νευρώνας, γίνεται υπολογισμός μιας συνάρτησης από το άθροισμα των γινομένων βαρών και εισόδων του. Έπειτα γίνεται σύγκριση της τιμής της συνάρτησης αυτής με την τιμή κατωφλίου, η οποία είναι χαρακτηριστική για τον κάθε νευρώνα. Στην περίπτωση που η τιμή της συνάρτησης είναι μεγαλύτερη της τιμής κατωφλίου, τότε υπολογίζεται η έξοδος. Στην συνέχεια, η έξοδος αυτή μπορεί να αποτελέσει είσοδο για τον επόμενο νευρώνα, ο οποίος αντίστοιχα θα παράγει μια έξοδο. Αναφορικά οι τιμές των βαρών είναι δυναμικές, αυτό παρατηρείται κατά την εκπαίδευση, όπου προσαρμόζονται σύμφωνα με το επιθυμητό αποτέλεσμα και τη μέθοδο εκπαίδευσης.

Η συνάρτηση που αναφέραμε παραπάνω αποκαλείται συνάρτηση ενεργοποίησης, η οποία ορίζει την έξοδο εν συνάρτησει του επιπέδου ενεργοποίησης της εισόδου. Διακρίνονται τρεις βασικοί τύποι:

1. **Συνάρτηση κατωφλίου**, από τον τύπο:

$$\varphi(v) = \begin{cases} 1, & \text{αν το } v \geq 0 \\ 0, & \text{αν το } v < 0 \end{cases}$$

Στην περίπτωση αυτή ο νευρώνας λειτουργεί ως δυαδικό στοιχείο με αποτέλεσμα η έξοδός του να παίρνει την τιμή 1 όταν είναι ενεργοποιημένος και την τιμή 0 όταν δεν είναι.

2. **Τμηματική Γραμμική Συνάρτηση**, από τον τύπο:

$$\varphi(v) = \begin{cases} 1, & \text{όταν } v \geq \frac{1}{2} \\ v, & \text{όταν } -\frac{1}{2} < v < \frac{1}{2} \\ 0, & \text{όταν } v \leq -\frac{1}{2} \end{cases}$$

3. **Σιγμοειδής Συνάρτηση**, από τον τύπο:

$$\varphi(v) = \frac{1}{1+e^{-v}}$$

Πρόκειται για την συνάρτηση που χρησιμοποιείται πιο συχνά για την κατασκευή ΤΝΔ. Στην ουσία πρόκειται για μια αύξουσα συνάρτηση, που δίνεται από τον παραπάνω τύπο.

Οι δύο τελευταίες συναρτήσεις είναι αυτές που συναντάμε πιο συχνά ως συναρτήσεις ενεργοποίησης (ή συνάρτηση μεταφοράς).



### 1.3.2.2. Δομή Τεχνητών Νευρωνικών Δικτύων

Ένα ΤΝΔ αποτελείται από πολλούς νευρώνες διασυνδεδεμένους μεταξύ τους, οι οποίοι αλλιώς καλούνται υπολογιστικοί κόμβοι. Οι κόμβοι αυτοί είναι δομημένοι σε στρώματα, τα οποία ονομάζονται και επίπεδα (layers). Τα σημεία εισόδου που δέχονται τα δεδομένα αποκαλούνται μονάδες εισόδου και αποτελούν το αντίστοιχο επίπεδο εισόδου (input layer). Το επίπεδο το οποίο μεταφέρει δεδομένα εκτός του δικτύου αποκαλείται επίπεδο εξόδου (output layer). Τα εσωτερικά επίπεδα, στα οποία γίνεται η ενδιάμεση εσωτερική επεξεργασία, ονομάζονται κρυφά επίπεδα (hidden layers). Δεν είναι υποχρεωτικό κάθε Νευρωνικό Δίκτυο να περιέχει κρυφά επίπεδα. Οι μονάδες εισόδου είναι ίσες με τις μεταβλητές εισόδου και αντίστοιχα οι νευρώνες εξόδου είναι ίσοι με τις μεταβλητές εξόδου, όμως τα κρυφά επίπεδα ενδέχεται να έχουν οποιοδήποτε αριθμό νευρώνων.

### 1.3.2.3. Αρχιτεκτονική Τεχνητών Νευρωνικών Δικτύων

Η δομή των νευρώνων ενός ΤΝΔ εξαρτάται από το αλγόριθμο εκμάθησης για την εκπαίδευση του δικτύου. Τα δίκτυα στα οποία τα σήματα πηγαινούν από την είσοδο προς την έξοδο, χαρακτηρίζονται ως δίκτυα εμπρός τροφοδότησης. Όταν η έξοδος ενός νευρώνα είναι είσοδος κάποιου άλλου νευρώνα τότε έχουμε την περίπτωση της ανάδρασης.

Διακρίνονται οι εξής περιπτώσεις δομών:

1. **Εμπρός Τροφοδότησης – Ενός Επιπέδου Δίκτυα:** Οι νευρώνες είναι δομημένοι σε επίπεδα.
2. **Εμπρός Τροφοδότησης – Πολλών Επιπέδων Δίκτυα:** Υπάρχουν παραπάνω από ένα κρυφά επίπεδα, οι κόμβοι των οποίων καλούνται κρυφοί νευρώνες. Στο κάθε επίπεδο οι νευρώνες παίρνουν σαν είσοδο την έξοδο του προηγούμενου επιπέδου.
3. **Αναδρομικά Δίκτυα:** Χαρακτηρίζονται από το γεγονός της ύπαρξης τουλάχιστον ενός κόμβου ανάδρασης.

#### 1.3.2.4. Πλεονεκτήματα και Μειονεκτήματα Τεχνητών Νευρωνικών Δικτύων

Ο κύριος στόχος των Νευρωνικών δικτύων είναι να καταφέρει να υλοποιήσει τις ίδιες γνωστικές λειτουργίες που εκτελεί και ο ανθρώπινος νους, όπως: η δεξιότητα αναγνώρισης προτύπων, η επίλυση προβλημάτων, η μάθηση και άλλα. Αυτό το πλήθος λειτουργιών μπορεί να εφαρμοστεί σε ένα ευρύ φάσμα, όπως είναι η χρησιμοποίηση τους για τη πρόβλεψη του αποτελέσματος ή η εύρεση της σωστής απόφασης μέσω συγκεκριμένων διαδικασιών. Σε περίπτωση όμως που στα προβλήματα που έχουν να αντιμετωπίσουν δεν είναι δυνατή η πρόβλεψη του αποτελέσματος, τότε χρησιμοποιούνται οι ακόλουθες λύσεις προβλημάτων: η ταξινόμηση (classification), η αναγνώριση (recognition), η αποτίμηση (assessment) και η πρόβλεψη (prediction).

Ένας τέτοιος τομέας ο οποίος είναι σχετικά πρόσφατος στον κλάδο της τεχνολογίας και με τη ραγδαία εξέλιξη που έχει, φέρει μερικά πλεονεκτήματα που προσδιορίζουν τα Νευρωνικά Δίκτυα, κάποια από αυτά είναι:

- Προσαρμοστικότητα: Τα βάρη προσαρμόζονται σύμφωνα με τις αλλαγές του περιβάλλοντος τους.
- Αποδεικτική απόκριση: Η ικανότητα του δικτύου να παρέχει πληροφορίες όχι μόνο για το συγκεκριμένο παράδειγμα που έχουμε αλλά και για την εμπιστοσύνη ότι η σωστή απόφαση θα ληφθεί.
- Αντοχή σε σφάλματα: Ένα δίκτυο το οποίο είναι υλοποιημένο σε hardware έχει ανεκτικότητα σε σφάλματα εάν ληφθούν οι βασικές πολιτικές ασφάλειας ενός συστήματος, η προστασία του δεν διαφοροποιείται (για το υλικό), λόγω των χαρακτηριστικών που διακρίνουν ένα Νευρωνικό Δίκτυο.
- Γενικό μοντέλο: Μπορούμε να αναπαραστήσουμε σχεδόν οτιδήποτε επιθυμούμε.
- Χρησιμοποιούνται για: γραμμική παρεμβολή, μη γραμμική παρεμβολή και ταξινόμηση.
- Συναντάται ομοιομορφία στην ανάλυση και το σχεδιασμό στα περισσότερα πεδία που εφαρμόζεται, διότι υπάρχει ο ίδιος συμβολισμός.

Βέβαια πέρα από τα θετικά αυτά χαρακτηριστικά παρατηρούνται και μερικά αρνητικά χαρακτηριστικά, κάποια από τα οποία είναι τα εξής:

- Χρειάζεται να γίνει βελτιστοποίηση για να βρούμε τις παραμέτρους.
- Δεν υπάρχει τόσο καλή γενίκευση.
- Είναι δύσκολη η επεξήγηση της γνώσης (ποιοτικά) που μοντελοποιούν.
- Υπάρχει ανάγκη μάθησης και αυτό απαιτεί μεγάλο αριθμό δεδομένων για να γίνουν αρκετές δοκιμές, ώστε να είναι βέβαιο το σωστό αποτέλεσμα.

Υπάρχουν αρκετοί τύποι Νευρωνικών δικτύων, με διαφορετικά θετικά και αρνητικά χαρακτηριστικά. Μερικοί από αυτούς που παρουσιάζουν ενδιαφέρον είναι τα Επαναλαμβανόμενα Νευρωνικά Δίκτυα (RNN). Τα RNN είναι ιδιαίτερα ισχυρά όταν πρόκειται για επίλυση εργασιών νευρογλωσσικού προγραμματισμού, είναι ήδη επιτυχημένα σε προβλήματα αναγνώρισης φωνής και μηχανικής μετάφρασης. Ένας ακόμη τύπος είναι τα Συνελκτικά Νευρωνικά Δίκτυα (CNN), τα οποία χρησιμοποιούνται σε μεγάλο ποσοστό στην αναγνώριση εικόνας. Πιο αναλυτικά θα αναφερθούν παρακάτω.

## 1.4. Βαθιά Μάθηση (Deep Learning) – Τι είναι και πως λειτουργεί στα ΤΝΔ;

### 1.4.1 Τι είναι η Βαθιά Μάθηση

Η Βαθιά Μάθηση[9]–[11] είναι ένα υποσύνολο της Μηχανικής Μάθησης, πρόκειται για ένα Νευρωνικό Δίκτυο το οποίο αποτελείται από τουλάχιστον τρία επίπεδα. Τα Νευρωνικά Δίκτυα προσπαθούν να μιμηθούν την συμπεριφορά του ανθρώπινου εγκεφάλου, επιτρέποντας σε αυτά να μάθουν μέσα από έναν μεγάλο όγκο δεδομένων. Τα Νευρωνικά Δίκτυα ενός μόνο επιπέδου μπορούν να κάνουν προβλέψεις κατά προσέγγιση, η ύπαρξη πρόσθετων κρυφών επιπέδων μπορεί να βοηθήσει στη βελτιστοποίηση για μεγαλύτερη ακρίβεια.

Αποτελεί κινητήριος δύναμη πολλών εφαρμογών και υπηρεσιών Τεχνητής Νοημοσύνης, τα οποία βελτιώνουν την αυτοματοποίηση και πραγματοποιούν αναλυτικές και φυσικές εργασίες χωρίς την παρέμβαση του ανθρώπου. Ορισμένα παραδείγματα της εμφάνισης της στην καθημερινή ζωή των ανθρώπων είναι τα εξής: ψηφιακοί βοηθοί, τηλεκοντρόλ με λειτουργία φωνής κλπ. Καθώς επίσης έχει εφαρμογή και σε αναδυόμενες τεχνολογίες, όπως είναι τα αυτοοδηγούμενα αυτοκίνητα (π.χ. Tesla).

### 1.4.2. Deep Learning vs Machine Learning

Αρχικά αξίζει να σημειωθούν ορισμένες διαφορές ανάμεσα στη Βαθιά Μάθηση και στη Μηχανική Μάθηση[12]–[14]. Η Βαθιά Μάθηση διαφοροποιείται από τη Μηχανική από τον τύπο των δεδομένων τα οποία επεξεργάζεται και από τις μεθόδους της.

Οι αλγόριθμοι της Μηχανικής Μάθησης αξιοποιούν δομημένα επισημασμένα σύνολα δεδομένων για να κάνουν προβλέψεις, αυτό σημαίνει ότι συγκεκριμένα χαρακτηριστικά ορίζονται από τα δεδομένα εισόδου για το μοντέλο και οργανώνονται σε πίνακες. Αυτό δεν σημαίνει απόλυτα ότι δεν χρησιμοποιεί μη δομημένα δεδομένα, αλλά αν χρησιμοποιήσει εκτελεί κάποια προ-επεξεργασία έτσι ώστε να τα οργανώσει σε αντίστοιχη δομημένη μορφή.

Η Βαθιά Μάθηση καταργεί κάποια από τα προ-επεξεργασίας δεδομένα, τα οποία έχουν να κάνουν με την Μηχανική Μάθηση. Οι αλγόριθμοι της Βαθιάς Μάθησης επεξεργάζονται μη δομημένα δεδομένα, όπως κείμενα και εικόνες, επίσης αυτοματοποιούν την εξαγωγή χαρακτηριστικών, αφαιρώντας κατά ένα μέρος την εξάρτησή τους από τον άνθρωπο. Για παράδειγμα: Υπάρχουν φωτογραφίες από διάφορα ζώα και στόχος είναι η ταξινόμηση τους ανά κατηγορία π.χ. «σκύλος», «γάτα», «κουνέλι» κ.α. Οι αλγόριθμοι ΒΜ μπορούν να ορίσουν ποια χαρακτηριστικά είναι αυτά

που ξεχωρίζουν το κάθε ζώο π.χ. «αφτιά», «ουρά» κ.α. Ενώ στην MM αυτή η διαδικασία της ταξινόμησης χαρακτηριστικών γίνεται χειροκίνητα από τον άνθρωπο.

Έπειτα, μέσω των διαδικασιών «gradient descent» και «backpropagation», ο αλγόριθμος της BM προσαρμόζεται και αυτό του επιτρέπει να κάνει προβλέψεις για μια νέα φωτογραφία ζώου με μεγαλύτερη ακρίβεια.

Τα μοντέλα MM και BM έχουν διαφορετικούς τύπους μάθησης. Οι πιο σύνηθες κατηγορίες είναι οι εξής: Εποπτευόμενη Μάθηση, Μη Εποπτευόμενη Μάθηση και Ενισχυτική Μάθηση.

- Η Εποπτευόμενη Μάθηση χρησιμοποιεί επισημασμένα σύνολα δεδομένων για κατηγοριοποίηση ή για να κάνει προβλέψεις, αυτό απαιτεί την παρέμβαση του ανθρώπου έτσι ώστε να γίνει σωστά η εισαγωγή των δεδομένων.
- Σε αντίθεση, η Μη Εποπτευόμενη Μάθηση δεν απαιτεί επισημασμένα σύνολα δεδομένων, αντ' αυτού, εντοπίζει μοτίβα στα δεδομένα και τα ομαδοποιεί σύμφωνα με τα διακριτικά χαρακτηριστικά τους.
- Η Ενισχυτική Μάθηση είναι μια διαδικασία κατά την οποία ένα μοντέλο μαθαίνει να πετυχαίνει καλύτερη ακρίβεια, για την εκτέλεση μιας διεργασίας σε ένα περιβάλλον βασισμένο στην ανατροφοδότηση (feedback), έτσι ώστε να βελτιστοποιήσει την συνάρτηση ανταμοιβής (reward).

#### 1.4.3. Πως Λειτουργεί

Τα Τεχνητά Νευρωνικά Δίκτυα Βαθιάς Μάθησης προσπαθούν να μιμηθούν την σκέψη του ανθρώπινου εγκεφάλου μέσω: ενός συνδυασμού δεδομένων εισόδου, βαρών και εξόδου. Αυτά τα στοιχεία συνεργάζονται έτσι ώστε να αναγνωρίσουν, να κατηγοριοποιήσουν και να περιγράψουν αντικείμενα με ακρίβεια μέσα από τα δεδομένα.

Τα Τεχνητά Νευρωνικά Δίκτυα Βαθιάς Μάθησης αποτελούνται από πολλαπλά επίπεδα διασυνδεδεμένων κόμβων, το κάθε επίπεδο είναι βασισμένο στο προηγούμενο, με τον τρόπο αυτό βελτιώνει την πρόβλεψη ή την κατηγοριοποίηση. Η εξέλιξη των υπολογισμών μέσω του δικτύου ονομάζεται προς τα εμπρός διάδοση (forward propagation). Τα επίπεδα εισόδου και εξόδου ενός Τεχνητού Νευρωνικού Δικτύου Βαθιάς Μάθησης αποκαλούνται ορατά επίπεδα (visible layers).

Το επίπεδο εισόδου είναι το σημείο όπου το μοντέλο της Βαθιάς Μάθησης δέχεται τα δεδομένα που θα επεξεργαστεί και αντίστοιχα το επίπεδο εξόδου είναι το σημείο όπου γίνεται η τελική πρόβλεψη ή ταξινόμηση.

Μια άλλη διαδικασία η οποία αποκαλείται οπισθοδιάδοση (back propagation) χρησιμοποιεί αλγορίθμους, όπως είναι ο αλγόριθμος βελτιστοποίησης με επικλινή κάθοδο (gradient descent), για να υπολογίσει τα σφάλματα στις προβλέψεις και έπειτα να προσαρμόσει τα βάρη και τα σφάλματα του συστήματος της συνάρτησης, πηγαίνοντας προς τα πίσω μέσα από τα επίπεδα σε μια προσπάθεια να εκπαιδεύσει το μοντέλο. Η «Προς τα εμπρός διάδοση» (forward propagation) σε συνδυασμό με την «Οπισθοδιάδοση» (back propagation), επιτρέπουν σε ένα ΤΝΔ να κάνει προβλέψεις και να διορθώνει τυχόν σφάλματα. Με την πάροδο του χρόνου, ο αλγόριθμος αποκτά σταδιακά μεγαλύτερη ακρίβεια.

Τα παραπάνω αποτελούν τις πιο απλές μορφές ΤΝΔ Βαθιάς Μάθησης με τους απλούστερους όρους.

Όμως, κατά κύριο λόγο οι αλγόριθμοι Βαθιάς Μάθησης είναι ιδιαίτερα σύνθετοι. Υπάρχουν διαφορετικοί τύποι ΤΝΔ για την αντιμετώπιση συγκεκριμένων προβλημάτων. Ορισμένοι τύποι είναι οι εξής:

- Συνελκτικά Νευρωνικά Δίκτυα (Convolutional neural networks - CNNs): Χρησιμοποιούνται κυρίως σε εφαρμογές υπολογιστικής όρασης και ταξινόμησης εικόνων, όπως: στην ανίχνευση χαρακτηριστικών και μοτίβων μέσα σε μια εικόνα, στην υλοποίηση εργασιών (ανίχνευση ή αναγνώριση αντικειμένων).
- Επαναλαμβανόμενα Νευρωνικά Δίκτυα (Recurrent neural network - RNNs): Χρησιμοποιούνται κυρίως σε εφαρμογές φυσικής γλώσσας και αναγνώρισης ομιλίας, καθώς αξιοποιούν δεδομένα διαδοχικών ή χρονικών σειρών.

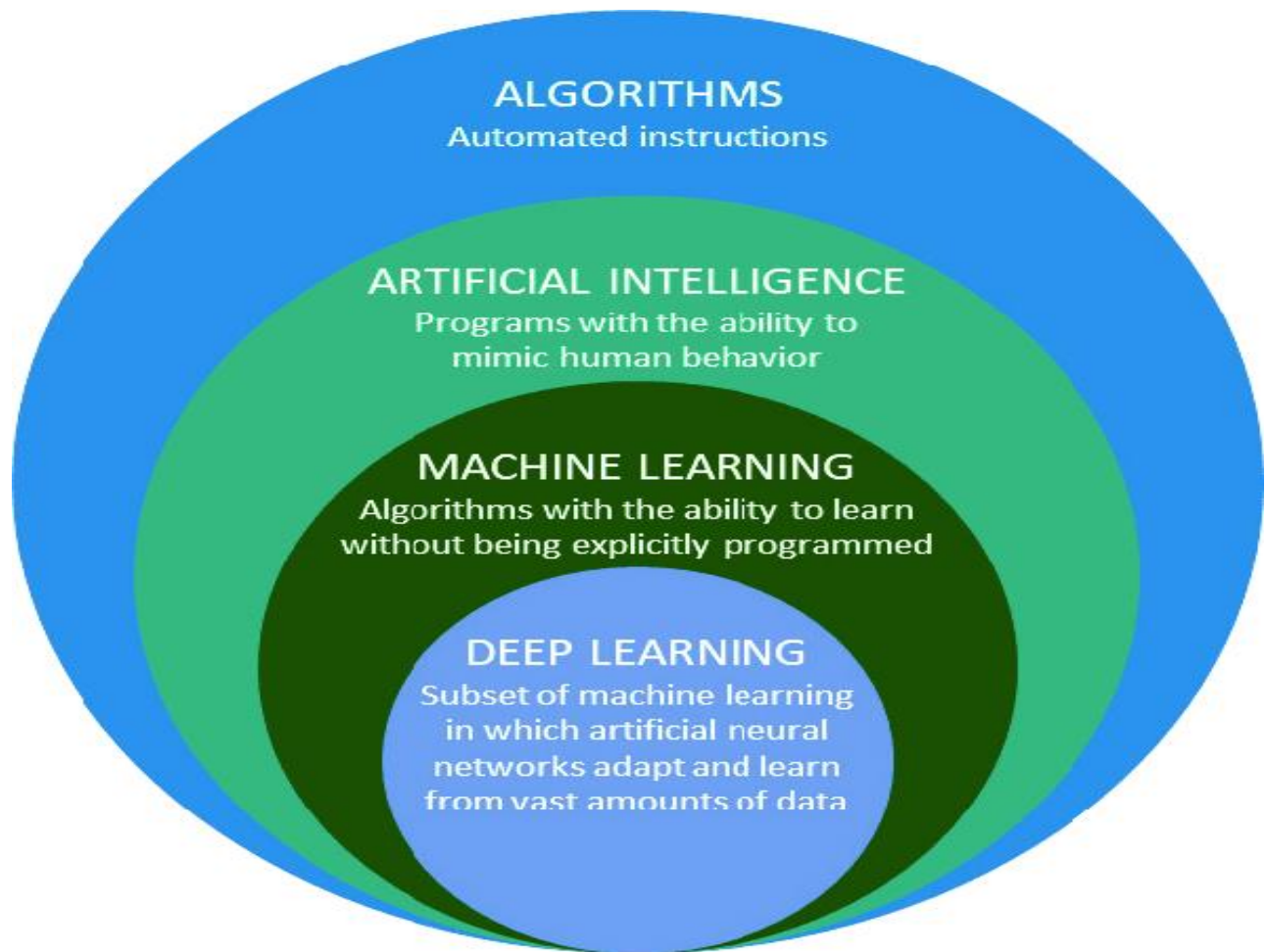
#### 1.4.4. Εφαρμογές της Βαθιάς Μάθησης

Οι εφαρμογές της Βαθιάς Μάθησης αποτελούν πλέον μέρος της καθημερινότητας μας, έχουν ενσωματωθεί σε τέτοιο βαθμό σε προϊόντα και υπηρεσίες που δεν γίνεται αντιληπτό από τους χρήστες ότι στο πίσω μέρος εκτελούνται περίπλοκες επεξεργασίες δεδομένων. Ορισμένα παραδείγματα εφαρμογών της είναι τα ακόλουθα:

- **Επιβολή του Νόμου:** Οι αλγόριθμοι μπορούν να αναλύσουν και να μάθουν από στοιχεία συναλλαγών, ώστε να ανιχνεύσουν επικίνδυνα μοτίβα, τα οποία υποδηλώνουν πιθανή απάτη ή εγκληματική δραστηριότητα. Επίσης, μπορούν να βελτιώσουν την αποτελεσματικότητα της έρευνας των αρχών, εξάγοντας στοιχεία από ηχογραφήσεις, βίντεο, φωτογραφίες, έγγραφα κ.α. Κάτι το οποίο αποδεικνύεται βοηθητικό στην ανάλυση μεγάλων ποσοτήτων δεδομένων, πιο γρήγορα και με μεγαλύτερη ακρίβεια.
- **Υπηρεσίες Εξυπηρέτησης Πελατών:** Η τεχνολογία της ΒΜ είναι ενσωματωμένη σε πολλούς οργανισμούς. Ένα χαρακτηριστικό παράδειγμα είναι τα λεγόμενα «chatbots», τα οποία συναντάμε σε μεγάλη ποικιλία εφαρμογών, σε υπηρεσίες, καθώς και σε πύλες εξυπηρέτησης πελατών. Πρόκειται για μια απλή μορφή Τεχνητής Νοημοσύνης (Artificial Intelligence - AI). Τα πιο απλά «chatbots» χρησιμοποιούν φυσική γλώσσα και σε ορισμένες περιπτώσεις οπτική αναγνώριση. Ενώ, τα πιο εξεζητημένα «chatbots» επιχειρούν να προσδιορίσουν (μέσω της μάθησης) αν υπάρχουν πολλαπλές απαντήσεις σε διαφορούμενες ερωτήσεις. Σύμφωνα με τις απαντήσεις που λαμβάνουν προσπαθούν να απαντήσουν απευθείας στα ερωτήματα ή να κατευθύνουν την συνομιλία σε χρήστη (άνθρωπο).

#### 1.4.5. Απαιτήσεις Hardware

Στην Βαθιά Μάθηση απαιτείται μεγάλη υπολογιστική δύναμη, για τον λόγο αυτό οι Κάρτες Γραφικών (Graphic Processor Units - GPUs) υψηλής απόδοσης είναι ιδανικές, καθώς μπορούν να χειριστούν μεγάλο όγκο υπολογισμών σε πολλαπλούς πυρήνες με αρκετή μνήμη RAM. Η διαχείριση όμως πολλαπλών GPUs είναι ιδιαίτερα δαπανηρή.



6. Δόμηση επιπέδων για την Αναγνώριση Προσώπου



## 1.5. Συνελκτικά Νευρωνικά Δίκτυα (Convolutional Neural Network - CNNs)

Τα Συνελκτικά Νευρωνικά Δίκτυα[15], [16] ξεχωρίζουν από άλλους τύπους Νευρωνικών Δικτύων λόγω της υψηλότερης απόδοσης των σημάτων εισόδου(εικόνας, ήχου η ομιλίας). Συναντάμε τρεις βασικούς τύπους επιπέδων:

1. Επίπεδο Συνέλιξης ή Συνελκτικό Επίπεδο (Convolutional Layer)
2. Επίπεδο Ομαδοποίησης/Συγκέντρωσης (Pooling Layer)
3. Πλήρως Συνδεδεμένο Επίπεδο (Fully – Connected Layer)

Το Συνελκτικό Επίπεδο είναι το πρώτο επίπεδο ενός Συνελκτικού Δικτύου. Στη συνέχεια, ακολουθούν είτε επιπρόσθετα Συνελκτικά Επίπεδα είτε Επίπεδα Ομαδοποίησης και τελευταίο είναι το Πλήρως Συνδεδεμένο Επίπεδο. Με κάθε επίπεδο, το ΣΝΔ γίνεται πολυπλοκότερο αναγνωρίζοντας μεγαλύτερα τμήματα της εικόνας. Τα αρχικά επίπεδα εστιάζουν σε απλούστερα χαρακτηριστικά, όπως είναι τα χρώματα και οι ακμές. Στα επόμενα επίπεδα γίνεται αναγνώριση μεγαλύτερων στοιχείων ή σχημάτων του αντικειμένου της εικόνας, έως ότου να αναγνωριστεί πλήρως το στοχευμένο αντικείμενο.

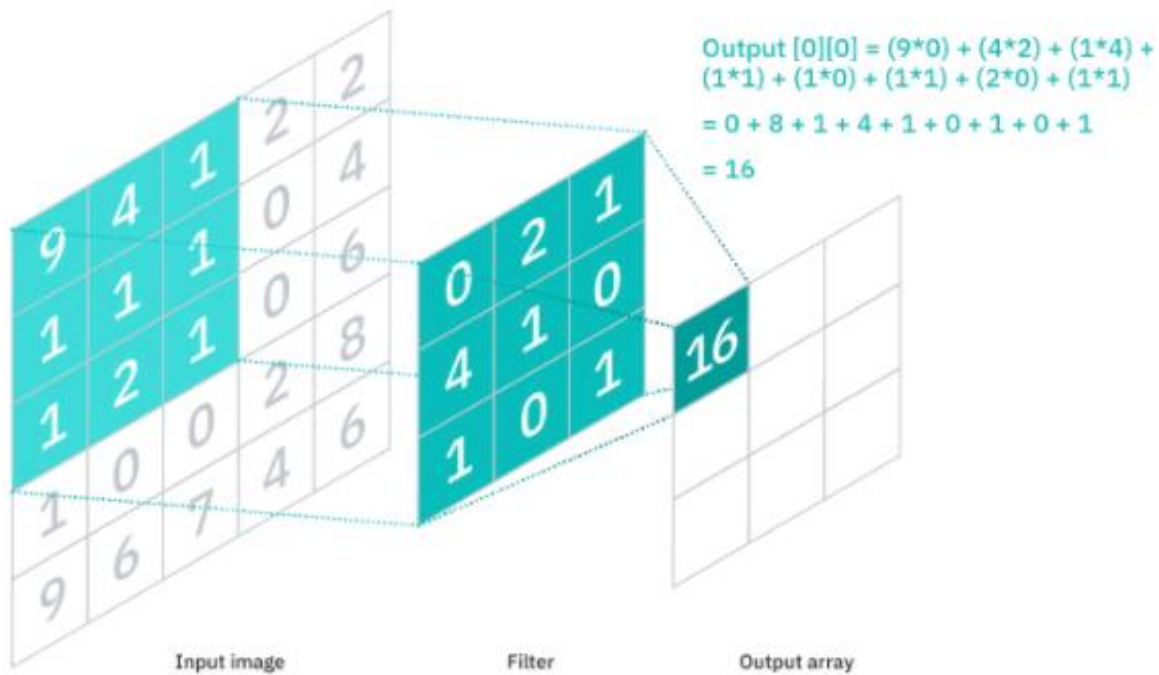
### Συνελκτικό Επίπεδο

Το Συνελκτικό Επίπεδο είναι το κύριο δομικό στοιχείο ενός ΣΝΔ, καθώς εκεί υλοποιούνται οι περισσότεροι υπολογισμοί. Αποτελείται από τρία βασικά στοιχεία: δεδομένα εισόδου, φίλτρο και χάρτης χαρακτηριστικών.

Έστω ότι έχουμε ως είσοδο μια έγχρωμη εικόνα, ή οποία αποτελείται από έναν τρισδιάστατο πίνακα εικονοστοιχείων (pixels). Δηλαδή, η είσοδος θα αποτελείται από τρεις διαστάσεις (ύψος, πλάτος, βάθος – height, width, depth), οι οποίες αντιστοιχούν στο πρότυπο χρώματος RGB (Red, Green, Blue) σε μια εικόνα. Επίσης έχουμε και έναν ανιχνευτή χαρακτηριστικών(πυρήνας ή φίλτρο), ο οποίος θα κινείται στα πεδία λήψης της εικόνας και θα ελέγχει αν υπάρχει το επιθυμητό χαρακτηριστικό. Η προ-αναφέρουσα διαδικασία αποκαλείται συνέλιξη.

Ο ανιχνευτής χαρακτηριστικών είναι ένας δισδιάστατος πίνακας βαρών, ο οποίος αναπαριστά ένα μέρος της εικόνας. Υπάρχει πληθώρα μεγθών των φίλτρων, αλλά συνήθως το μέγεθος του φίλτρου είναι ένας τρισδιάστατος πίνακας. Το μέγεθος του φίλτρου καθορίζει το μέγεθος των πεδίων λήψης.

Το φίλτρο εφαρμόζεται σε ένα τμήμα της εικόνας και υπολογίζεται το εσωτερικό γινόμενο μεταξύ των εικονοστοιχείων της εικόνας εισόδου και του φίλτρου. Έπειτα το εσωτερικό γινόμενο τροφοδοτείται σε έναν πίνακα εξόδου. Μετά, το φίλτρο μετακινείται με έναν βηματισμό, επαναλαμβάνοντας της διαδικασία μέχρι το φίλτρο να έχει σαρώσει ολόκληρη την εικόνα. Η τελική έξοδος μιας σειράς εσωτερικών γινομένων μεταξύ της εισόδου και του φίλτρου καλείται χάρτης χαρακτηριστικών (αλλιώς χάρτης ενεργοποίησης ή συνελκτικό χαρακτηριστικό).



7. Παράδειγμα Συνελκτικού Επιπέδου

Σύμφωνα με την παραπάνω εικόνα, κάθε τιμή εξόδου στον χάρτη χαρακτηριστικών δεν συνδέεται στο κάθε εικονοστοιχείο (pixel) της εικόνας εισόδου, αλλά συνδέεται στο πεδίο λήψης όπου εφαρμόζεται το φίλτρο. Δεδομένου ότι ο πίνακας εξόδου δεν χρειάζεται να αντιστοιχεί κατευθείαν σε κάθε τιμή εισόδου (pixel), τα συνελκτικά επίπεδα και τα επίπεδα ομαδοποίησης αναφέρονται ως επίπεδα μερικής σύνδεσης (partially connected layers). Ωστόσο, αυτό το γνώρισμα μπορεί να περιγραφεί και ως τοπική σύνδεση (local connectivity).

Θα πρέπει να σημειωθεί ότι τα «βάρη» στον ανιχνευτή χαρακτηριστικών παραμένουν σταθερά καθώς αυτός κινείται πάνω στην εικόνα, αυτό το φαινόμενο καλείται «κοινή χρήση παραμέτρων» (parameter sharing). Ορισμένοι παράμετροι, όπως είναι η τιμές του βάρους, προσαρμόζονται κατά

την διάρκεια της εκπαίδευσης μέσω των διαδικασιών: της «οπισθοδιάδοσης» (backpropagation) και του αλγορίθμου βελτιστοποίησης με επικλινή κάθοδο (gradient descent). Υπάρχουν τρεις παράμετροι που επηρεάζουν το μέγεθος του όγκου της εξόδου, οι οποίοι θα πρέπει να οριστούν πριν ξεκινήσει η διαδικασία της εκπαίδευσης του Νευρωνικού Δικτύου. Αυτές είναι:

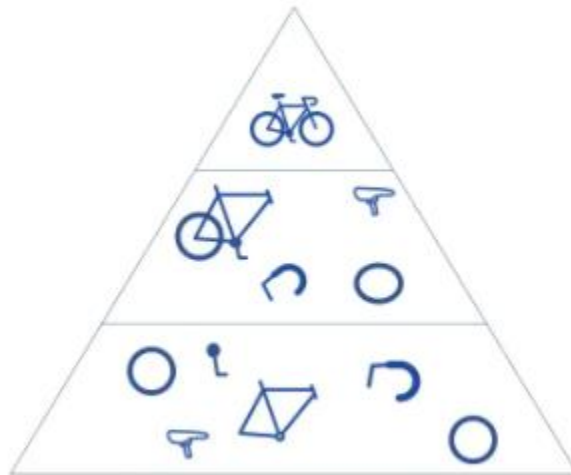
1. Αριθμός των Φίλτρων: ο οποίος επηρεάζει το βάθος (depth) της εξόδου. Τρία διαφορετικά φίλτρα αποδίδουν τρεις διαφορετικούς χάρτες χαρακτηριστικών, δημιουργώντας ένα βάθος των τριών.
2. Διασκελισμός (stride): είναι η απόσταση, ή αλλιώς αριθμός των εικονοστοιχείων, όπου το φίλτρο κινείται πάνω στον πίνακα εισόδου. Η τιμή διασκελισμού ίση με 2 ή μεγαλύτερο του 2 (είναι σπάνια). Μεγάλη τιμή διασκελισμού αποδίδει μικρότερη τιμή εξόδου.
3. Μηδενικό-Περιθώριο (Zero - Padding): χρησιμοποιείται όταν το φίλτρο δεν χωράει στην εικόνα εισόδου. Θέτει όλα τα στοιχεία που βρίσκονται εκτός του πίνακα εισόδου στο μηδέν, παράγοντας έτσι μια μεγαλύτερη ή ίσου μεγέθους έξοδο. Υπάρχουν τρία είδη περιθωρίου:
  - a. Έγκυρο Περιθώριο (Valid - Padding ): ονομάζεται και «χωρίς-περιθώριο». Η τελευταία συνέλιξη απορρίπτεται εάν οι διαστάσεις δεν συμπίπτουν.
  - b. Ίδιο Περιθώριο (Same - Padding): διαβεβαιώνει ότι το επίπεδο εξόδου έχει το ίδιο μέγεθος με το επίπεδο εισόδου.
  - c. Πλήρες Περιθώριο (Full - Padding): αυξάνει το μέγεθος της εξόδου εισάγοντας μηδενικά(0) στο περίγραμμα της εισόδου.

Ύστερα από κάθε συνελκτική διαδικασία, ένα ΣΝΔ εκτελεί έναν μετασχηματισμό Γραμμικής Μονάδας Διόρθωσης (Rectified Linear Unit - ReLU) στο χάρτη χαρακτηριστικών, εισάγοντας τη Μη-Γραμμικότητα στο μοντέλο.

Προηγουμένως αναφέραμε ότι μετά από το αρχικό επίπεδο συνέλιξης μπορεί να ακολουθήσει ένα άλλο επίπεδο συνέλιξης. Όταν συμβαίνει αυτό η δομή του ΣΝΔ μπορεί να γίνει ιεραρχική, καθώς τα επόμενα επίπεδα μπορούν να δουν τα εικονοστοιχεία μέσα στα πεδία λήψης των προηγούμενων επιπέδων.

Ας υποθέσουμε, ότι προσπαθούμε να προσδιορίσουμε αν μια εικόνα περιέχει ένα ποδήλατο. Το ποδήλατο είναι ένα «άθροισμα» από εξαρτήματα, αποτελείται από: τον σκελετό, το τιμόνι, τις

ρόδες, τα πετάλια κ.α. Κάθε μεμονωμένο εξάρτημα του ποδηλάτου δημιουργεί ένα μοτίβο χαμηλού επιπέδου στο Νευρωνικό Δίκτυο και ο συνδυασμός των εξαρτημάτων αντιπροσωπεύει ένα μοτίβο υψηλού επιπέδου, δημιουργώντας έτσι μια ιεραρχία χαρακτηριστικών μέσα στο ΣΝΔ. Ουσιαστικά, το συνελκτικό επίπεδο μετατρέπει την εικόνα σε αριθμητικές τιμές, επιτρέποντας έτσι στο Νευρωνικό Δίκτυο να ερμηνεύει και να εξάγει σχετικά μοτίβα.



8. Συνελκτικό Επίπεδο - Παράδειγμα Ποδηλάτου

### **Επίπεδο Ομαδοποίησης**

Τα Επίπεδα Ομαδοποίησης είναι επίσης γνωστά και ως επίπεδα «κατώτερης-δειγματοληψίας» (downsampling), διεξάγουν διαστασιακή μείωση, μειώνοντας τον αριθμό των παραμέτρων στην είσοδο. Όπως και στο Συνελκτικό Επίπεδο, κατά τη διαδικασία της Ομαδοποίησης ένα φίλτρο σαρώνει ολόκληρη την είσοδο, με μόνη διαφορά ότι το φίλτρο δεν έχει βάρη. Αντ' αυτού, το φίλτρο εφαρμόζει μια συνάρτηση συνάθροισης στις τιμές του πεδίου λήψης, γεμίζοντας τον πίνακα εξόδου. Υπάρχουν δυο βασικοί τύποι Ομαδοποίησης, οι οποίοι είναι:

1. Μέγιστης Ομαδοποίησης (Max Pooling): Καθώς το φίλτρο μετακινείται κατά μήκος της εισόδου, επιλέγει τα εικονοστοιχεία με τη μεγαλύτερη τιμή, έτσι ώστε να σταλούν στον πίνακα εξόδου. Η μέθοδος αυτή χρησιμοποιείται πιο συχνά από την μέθοδο Μέσης Ομαδοποίησης.

2. Μέσης Ομαδοποίησης (Average Pooling): Καθώς το φίλτρο μετακινείται κατά μήκος της εισόδου, υπολογίζει τη μέση τιμή του πεδίου λήψης, έτσι ώστε να σταλεί στον πίνακα εξόδου.

Παρά την απώλεια πληροφοριών στο Επίπεδο Ομαδοποίησης, υπάρχουν πολλά οφέλη για το ΣΝΔ. Ορισμένα από τα οφέλη αυτά, είναι: μείωση πολυπλοκότητας, βελτίωση της αποδοτικότητας και περιορισμός του ρίσκου της υπερβολικής-προσαρμογής (overfitting).

### **Πλήρως Συνδεδεμένο Επίπεδο**

Όπως προ αναφέρθηκε, οι τιμές των εικονοστοιχείων της εικόνας εισόδου δεν είναι απευθείας συνδεδεμένες με το επίπεδο εξόδου, στα μερικώς συνδεδεμένα επίπεδα. Παρόλα αυτά, στο Πλήρως Συνδεδεμένο Επίπεδο κάθε κόμβος στο επίπεδο εξόδου συνδέεται απευθείας σε έναν κόμβο του προηγούμενου επιπέδου.

Το επίπεδο αυτό εκτελεί τη διαδικασία της ταξινόμησης (classification) με βάση τα εξαγόμενα χαρακτηριστικά μέσω των προηγούμενων επιπέδων και των διαφορετικών φίλτρων τους. Ενώ τα Επίπεδα Συνέλιξης και Ομαδοποίησης τείνουν να κάνουν χρήση συναρτήσεων Γραμμικής Μονάδας Διόρθωσης (ReLU), τα Πλήρως Συνδεδεμένα Επίπεδα χρησιμοποιούν συνήθως μια κανονικοποιημένη εκθετική συνάρτηση (softmax activation function) για να ταξινομήσουν κατάλληλα τις εισόδους, παράγοντας μια πιθανότητα με τιμές από «0» έως «1».

## 1.6. Αναγνώριση Προσώπου

Η Αναγνώριση Προσώπου είναι ένας τρόπος αναγνώρισης του ανθρώπινου προσώπου με την βοήθεια της τεχνολογίας. Ένα σύστημα Αναγνώρισης Προσώπου χρησιμοποιεί βιομετρικά στοιχεία για να χαρτογραφήσει τα χαρακτηριστικά του προσώπου από μια φωτογραφία ή βίντεο. Έπειτα συγκρίνει τις πληροφορίες με μια βάση δεδομένων, ήδη γνωστών προσώπων, έτσι ώστε να βρει το αντίστοιχο ταίρι. Η Αναγνώριση Προσώπου μπορεί να βοηθήσει στην επιβεβαίωση της ταυτότητας ενός ανθρώπου, αλλά την ίδια στιγμή προβάλλει ζητήματα απορρήτου.

### 1.6.1. Πως λειτουργεί

Για τους περισσότερους ανθρώπους είναι εύκολο να αναγνωρίζουν τα πρόσωπα των μελών της οικογένειάς τους, των φίλων ή γνωστών τους, καθώς είναι εξοικειωμένοι με τα χαρακτηριστικά των προσώπων τους (όπως π.χ. μύτη, μάτια, στόμα κ.α.).

Με τον ίδιο τρόπο λειτουργεί και ένα σύστημα αναγνώρισης προσώπου, αλλά σε μεγαλύτερη αλγοριθμική κλίμακα. Εκεί όπου ο άνθρωπος βλέπει ένα πρόσωπο, η τεχνολογία Αναγνώρισης Προσώπου βλέπει δεδομένα. Αυτά τα δεδομένα μπορούν να αποθηκευτούν και να προσπελαστούν.

Σύμφωνα με την μελέτη του Πανεπιστημίου Τζορτζ-Τάουν (University of Georgetown), οι εικόνες/φωτογραφίες του μισού Αμερικάνικου πληθυσμού ενηλίκων είναι αποθηκευμένες σε μία ή περισσότερες βάσεις δεδομένων Αναγνώρισης Προσώπου, στις οποίες μπορούν να έχουν πρόσβαση οι υπηρεσίες αμερικάνικων αρχών.

Αν και οι τεχνολογίες της Αναγνώρισης Προσώπου ποικίλουν, υπάρχουν κάποια βασικά βήματα, τα οποία είναι:

- Βήμα 1. Η εικόνα του επιθυμητού προσώπου αποθανατίζεται με την μορφή αρχείου εικόνας ή βίντεο. Δεν υπάρχει περιορισμός για το αν το άτομο εμφανίζεται στην εικόνα μόνο του ή μέσα σε πλήθος, αν κοιτάει στην ευθεία ή πλάγια.
- Βήμα 2. Το λογισμικό της Αναγνώρισης Προσώπου διαβάζει τη γεωμετρία του επιθυμητού προσώπου. Βασικοί παράγοντες είναι η απόσταση μεταξύ των ματιών και η απόσταση μεταξύ μετώπου και πηγουνιού. Το λογισμικό εντοπίζει τα χαρακτηριστικά του προσώπου, τα οποία είναι απαραίτητα για τη διάκριση του

προσώπου. Ενημερωτικά, ένα σύστημα εντοπίζει 68 από αυτά τα χαρακτηριστικά. Το αποτέλεσμα είναι η «υπογραφή» του προσώπου (facial signature).

Βήμα 3. Η «υπογραφή» του προσώπου (πρόκειται για έναν μαθηματικό τύπο) συγκρίνεται με μία βάση δεδομένων, η οποία αποτελείται από γνωστά πρόσωπα.

Βήμα 4. Τέλος, εξάγεται το αποτέλεσμα, δηλαδή αν η υπογραφή του επιθυμητού προσώπου ταυτίζεται με κάποια άλλη μέσα από τη βάση δεδομένων του συστήματος.

### 1.6.2. Ιστορική Αναδρομή

Τα πρώτα ίχνη της Αναγνώρισης Προσώπου εντοπίζονται την δεκαετία του '60. Την χρονική αυτή περίοδο ο μαθηματικός και επιστήμονας της πληροφορικής Woodrow Wilson Bledsoe ανέπτυξε πρώτος ένα σύστημα από μετρήσεις, το οποίο θα μπορούσε να χρησιμοποιηθεί για να τοποθετεί φωτογραφίες προσώπων σε διαφορετικές ταξινομήσεις. Για τον λόγο αυτό του έχει δοθεί ο τίτλος του «πατέρα» της τεχνολογίας της Αναγνώρισης Προσώπου.

Σύντομα οι υπηρεσίες των αμερικάνικων αρχών έδειξαν ιδιαίτερο ενδιαφέρον για την μελέτη του Bledsoe. Με αποτέλεσμα, από την δεκαετία του '70 έως την δεκαετία του '90, οι υπηρεσίες αυτές να έχουν αναπτύξει τα δικά τους συστήματα Αναγνώρισης Προσώπου. Τα συστήματα αυτά ήταν σε πρώιμο στάδιο σε σύγκριση με την σημερινή τεχνολογία, αλλά η εργασία σε αυτά τα συστήματα οδήγησε στην δημιουργία των σύγχρονων προγραμμάτων.

Το έτος 2001 οι αστυνομικές αρχές χρησιμοποίησαν την ΑΠ για να βοηθήσουν στην αναγνώριση ατόμων μέσα στο πλήθος κατά την διεξαγωγή του Super Bowl XXXV. Την ίδια χρονιά το Αστυνομικό Τμήμα της κομητείας Πινέλας στην Φλόριντα δημιούργησε τη δικιά του βάση δεδομένων ΑΠ.

Το 2010 οι υπολογιστές πλέον έχουν αποκτήσει αρκετά μεγαλύτερη υπολογιστική ισχύ, με αποτέλεσμα η ΑΠ να αρχίζει να γίνεται μια πιο συνηθισμένη λειτουργία.

Το 2011 χρησιμοποιήθηκε λογισμικό ΑΠ το οποίο επιβεβαίωσε την ταυτότητα του τρομοκράτη Οσάμα Μπιν Λάντεν.

Το 2015 το Αστυνομικό Τμήμα της Βαλτιμόρης χρησιμοποίησε την ΑΠ για την αναγνώριση των ατόμων που πήραν μέρος στις διαδηλώσεις μετά τον θάνατο του Φρέντι Γκρέι.

Πλέον οι καταναλωτές χρησιμοποιούν την ΑΠ μέσω των smartphones και άλλων προσωπικών συσκευών. Συγκεκριμένα το 2015, οι εφαρμογές «Windows Hello» και «Android's Trusted Face»

επέτρεψαν στους χρήστες να συνδεθούν στις συσκευές τους κοιτώντας απλώς στην κάμερα τους. Η Apple ανακοίνωσε την χρήση τεχνολογίας ΑΠ (Face ID) με το μοντέλο iPhone-X το 2017. Γενικώς υπάρχει μια αντιπαράθεση γύρω από την τεχνολογία της ΑΠ, με τους επικριτές να ισχυρίζονται ότι πρόκειται για παραβίαση της ιδιωτικής ζωής. Πόλεις όπως το Σαν Φρανσίσκο, το Όκλαντ και η Βοστώνη έχουν απαγορεύσει στις κυβερνήσεις να χρησιμοποιούν την ΑΠ. Έπειτα από τις διαδηλώσεις του «Black Lives Matter» ενάντια στην αστυνομική βία το καλοκαίρι του 2020, πολλοί τεχνολογικοί κολοσσοί (ορισμένοι από τους οποίους: IBM, Amazon και Microsoft), ανακοίνωσαν ότι από εδώ και πέρα δεν θα πωλούν την τεχνολογία της ΑΠ στις υπηρεσίες επιβολής του νόμου.

### 1.6.3 Αποτελέσματα Ερευνών

Σύμφωνα με δοκιμές που διεξάχθηκαν από το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας, έδειξαν ότι από τον Απρίλιο του 2020 ο καλύτερος αλγόριθμος ΑΠ είχε ποσοστό σφάλματος μόλις 0,08%. Πρόκειται για αισθητή βελτίωση σε σύγκριση με το ποσοστό σφάλματος 4,1% του 2014. Σύμφωνα με έρευνα του Κέντρου Στρατηγικών & Διεθνών Σπουδών (CSI) που διεξάχθηκε το ίδιο έτος, η ακρίβεια (accuracy) είναι μεγαλύτερη όταν οι αλγόριθμοι αναγνώρισης χρησιμοποιούνται παίρνοντας ως είσοδο καθαρές και στατικές εικόνες, όπως για παράδειγμα είναι οι φωτογραφίες ταυτότητας ή διαβατηρίου. Το αποτέλεσμα της έρευνας έδειξε ότι αν οι αλγόριθμοι της ΑΠ χρησιμοποιηθούν κατά αυτόν τον τρόπο, τότε μπορούν να πετύχουν ποσοστό ακρίβειας έως και 99,97%.

Στην πραγματικότητα όμως, τα ποσοστά της ακρίβειας είναι συνήθως χαμηλότερα από τα ποσοστά που προ αναφέρθηκαν. Τα αποτελέσματα της έρευνας του CSI έδειξαν ότι το ποσοστό σφάλματος ενός αλγορίθμου αυξάνεται από 0,1% στην περίπτωση που είσοδος είναι στατική εικόνα υψηλής ποιότητας, σε 9,3% όταν χρησιμοποιείται εικόνα που έχει τραβηχτεί μέσα σε πλήθος. Παρατηρήθηκε υψηλό ποσοστό σφαλμάτων, όταν τα άτομα δεν κοιτούσαν απευθείας στην κάμερα ή δεν ήταν ευδιάκριτα τα πρόσωπα τους λόγω σκιών, φωτισμού ή ύπαρξης άλλων αντικειμένων.

Μια άλλη πρόκληση για την ακρίβεια είναι η γήρανση. Αποδείχθηκε ότι οι αλγόριθμοι μέσης-βαθμίδας (middle-tier) είχαν αυξημένα ποσοστά σφάλματος κατά 10% όταν χρησιμοποιήθηκαν φωτογραφίες ατόμων οι οποίες είχαν τραβηχτεί δύο δεκαετίες πριν.



#### 1.6.4. Εφαρμογές στην καθημερινότητα

Ορισμένα παραδείγματα χρήσης της τεχνολογίας Αναγνώρισης Προσώπου τόσο από υπηρεσίες/οργανισμούς όσο και από ανθρώπους είναι τα εξής:

- Στα αεροδρόμια της κυβέρνησης των Ηνωμένων Πολιτειών. Συστήματα ΑΠ ελέγχουν τους ανθρώπους οι οποίοι εισέρχονται και εξέρχονται στα αεροδρόμια. Το Τμήμα Εσωτερικής Ασφάλειας χρησιμοποιεί την τεχνολογία αυτή για να εντοπιστούν άτομα τα οποία είτε έχουν υπερβεί το όριο της παραμονής τους στην χώρα, είτε βρίσκονται υπό κατηγορίας κακουργήματος. Χαρακτηριστικό παράδειγμα, το 2018 στο Διεθνές Αεροδρόμιο της Ουάσιγκτον με την βοήθεια της ΑΠ έγινε σύλληψη ενός εγκληματία ο οποίος προσπαθούσε να εισέλθει παράνομα στην χώρα.
- Από εταιρίες κατασκευής κινητών τηλεφώνων στα προϊόντα τους. Η Apple ήταν η πρώτη εταιρία που χρησιμοποίησε την ΑΠ, αρχικά ως μέθοδος ξεκλειδώματος του μοντέλου iPhone X, κάτι το οποίο διατήρησε και σε επόμενα μοντέλα. Η μέθοδος αυτή είναι γνωστή ως «Face ID» και πιστοποιεί την ταυτότητα του κατόχου ώστε να ξεκλειδώσει η συσκευή. Η εταιρία ισχυρίζεται ότι η πιθανότητα να μπορέσει κάποιος άλλος άνθρωπος εκτός του κατόχου να ξεκλειδώσει την συσκευή είναι μια στο εκατομμύριο.
- Σε πλατφόρμες κοινωνικής δικτύωσης. Χαρακτηριστικό παράδειγμα αποτελεί το Facebook, το οποίο χρησιμοποιώντας έναν ειδικό αλγόριθμο εντοπίζει πρόσωπα, όταν οι χρήστες ανεβάζουν φωτογραφίες στην πλατφόρμα. Αφού ανέβει η εικόνα/εικόνες, η πλατφόρμα ρωτάει τους χρήστες αν συμφωνούν να προσθέσουν με την μορφή ετικέτας «tag» τα άτομα που απεικονίζονται στην φωτογραφία/φωτογραφίες. Στην περίπτωση που οι χρήστες αποδεχθούν, δημιουργείται ένας σύνδεσμος στο προφίλ των ατόμων αυτών. Το Facebook ισχυρίζεται ότι μπορεί να αναγνωρίσει πρόσωπα με ποσοστό ακρίβειας 98%.
- Από επιχειρήσεις για πρόσβαση των εργαζομένων στις εισόδους και στις περιοχές περιορισμένης πρόσβασης: Πολλές εταιρίες έχουν αντικαταστήσει τις κοινές κάρτες ασφαλείας με συστήματα ΑΠ.

### 1.6.5. Πλεονεκτήματα και Μειονεκτήματα

Ως μια σχετικά καινούργια τεχνολογία, δεν έχουν γίνει πλήρως κατανοητά τα πλεονεκτήματα και τα μειονεκτήματά της. Παρόλα αυτά μια γενική ιδέα θετικών και αρνητικών της αναφέρονται παρακάτω.

#### **Πλεονεκτήματα**

- Εντοπισμός αγνοουμένων ανθρώπων: Με την βοήθεια της ΑΠ οι υπηρεσίες επιβολής του νόμου καταφέρουν να εντοπίζουν εξαφανισμένα άτομα, ορισμένες φορές ακόμη και μετά από το πέρασμα πολλών ετών.
- Αναγνώριση εγκληματιών: Η ΑΠ βοηθά τις υπηρεσίες επιβολής του νόμου στον εντοπισμό εγκληματιών ή υπόπτων για πιθανά εγκλήματα.
- Ασφάλεια στις αεροπορικές πτήσεις: Πλέον σχεδόν όλα τα αεροδρόμια παγκοσμίως χρησιμοποιούν την ΑΠ για τον εντοπισμό εγκληματιών και για την αποφυγή πιθανών απειλών κατά την είσοδο αυτών στον χώρο του αεροδρομίου ή κατά την προσπάθεια αυτών να επιβιβαστούν στις πτήσεις.

#### **Μειονεκτήματα**

- Απειλείται η προσωπική ιδιωτικότητα του καθενός: Μεγάλη ανησυχία δημιουργείται λόγω της αποθήκευσης των προσώπων (φωτογραφιών) των ατόμων σε βάσεις δεδομένων στις οποίες μπορούν να έχουν πρόσβαση υπηρεσίες και οργανισμοί.
- Λανθασμένη ταυτοποίηση: Η τεχνολογία δεν είναι αλάνθαστη, πάντα θα υπάρχει έστω και ένα πολύ μικρό ποσοστό σφάλματος. Με αποτέλεσμα αν συμβεί αυτό το σφάλμα να κατηγορηθεί άδικα κάποιος αθώος πολίτης από τις υπηρεσίες επιβολής του νόμου.
- Παραπλάνηση των συστημάτων ΑΠ: Οι εγκληματίες βρίσκουν διάφορους τρόπους ώστε να ξεγελάσουν τα συστήματα φορώντας μάσκες ή μεταμφιέσεις, μειώνοντας την αποτελεσματικότητα της τεχνολογίας.
- Η γήρανση των ατόμων μειώνει την αποδοτικότητα της: Έρευνες έχουν αποδείξει ότι καθώς οι άνθρωποι γερνούν, τα χαρακτηριστικά τους αλλοιώνονται, με αποτέλεσμα να αυξάνεται η δυσκολία αναγνώρισης τους από το σύστημα ΑΠ. Μια άλλη έρευνα δείχνει

ότι η τεχνολογία αυτή είναι λιγότερο αποτελεσματική στο να αναγνωρίσει έγχρωμα άτομα και γυναίκες.

### 1.7. Open-Source Πλατφόρμες Αναγνώρισης Προσώπου

Τα συστήματα Αναγνώρισης Προσώπου σε Πραγματικό Χρόνο είναι ένα φλέγον θέμα στην όραση υπολογιστών (computer vision), για τον λόγο αυτό πολλές εταιρίες έχουν αναπτύξει τις δικές τους λύσεις (λογισμικό) έτσι ώστε να εκμεταλλευτούν την αναπτυσσόμενη αγορά. Σε σύγκριση με τις παραδοσιακές μεθόδους αναγνώρισης, τα συστήματα αναγνώρισης προσώπου σε πραγματικό χρόνο έχουν το πλεονέκτημα της χρήσης πολλαπλών περιπτώσεων (χρονικών στιγμών) του ίδιου ατόμου σε διαδοχικά frames.

Τα λογισμικά ανοιχτού κώδικα (open-source softwares) έχουν πολλά πλεονεκτήματα. Αρχικά, με τον ανοιχτό κώδικα, ο χρήστης καθορίζει τον τρόπο επεξεργασίας των δεδομένων. Δεύτερον, οι εργασίες ανοιχτού κώδικα (open-source projects) τείνουν να είναι καλύτερης ποιότητας. Αυτό συμβαίνει διότι γίνεται διαρκώς αναθεώρηση του κώδικα από πολλούς προγραμματιστές, με αποτέλεσμα τα «λάθη» (errors) να ανιχνεύονται σε σύντομο χρονικό διάστημα. Τρίτον, λόγω των χαμηλών τελών αδειοδότησης (licensing fees), τέτοιου είδους projects αναπτύσσονται εσωτερικά από μεμονωμένα άτομα ή από έναν διακριτικό πάροχο υπηρεσιών τεχνολογίας διαδικτύου. Είναι δύσκολο έως απίθανο να βρεθεί «ξεπερασμένο» λογισμικό ανοιχτού κώδικα, καθώς ακολουθεί σύγχρονες πρακτικές ανάπτυξης λογισμικού. Τέλος, ο ανοιχτός κώδικας θεωρείται ότι πρόκειται για το επόμενο επίπεδο εξέλιξης του κώδικα, καθώς δίνει στους προγραμματιστές την δυνατότητα να κατανοήσουν τον κώδικα πλήρως μέσα σε λίγα λεπτά.

Ύστερα από έρευνα στο αποθετήριο του «GitHub» για «Λογισμικά ανοιχτού κώδικα σε πραγματικό χρόνο για αναγνώριση προσώπου», γίνεται παρουσίαση μιας λίστας των καλύτερων επιλογών στην τρέχουσα χρονική περίοδο.

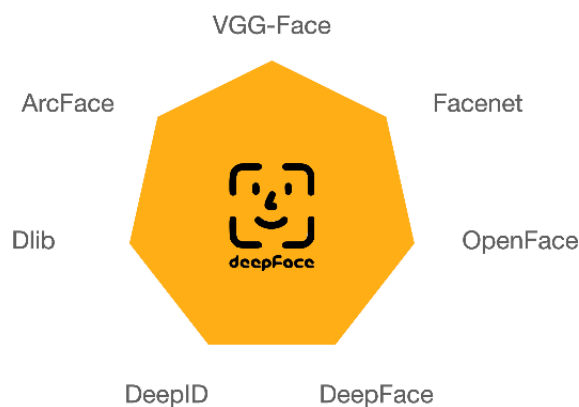
## 1. DeepFace



Αναπτύχθηκε από μια ομάδα ερευνητών του Facebook, πρόκειται για ένα «ελαφρύ» framework αναγνώρισης προσώπου και ανάλυσης χαρακτηριστικών προσώπου (ηλικία, φύλο, συναίσθημα κ.α.) για γλώσσα προγραμματισμού Python. Είναι ένα υβριδικό framework αναγνώρισης προσώπου καθώς περιλαμβάνει μοντέλα τελευταίας τεχνολογίας, όπως: VGG-Face, Google FaceNet, Facebook DeepFace, OpenFace, Dlib κ.α.

Πειράματα έχουν δείξει ότι ο άνθρωπος έχει ποσοστό ακρίβειας 97,53% σε εργασίες αναγνώρισης προσώπου, ενώ τα παραπάνω μοντέλα έχουν ήδη φτάσει και ξεπεράσει αυτό το ποσοστό ακρίβειας.

Επίσης, παρέχει ένα REST API, αλλά υποστηρίζει μόνο μεθόδους επικύρωσης, αυτό σημαίνει ότι η δημιουργία συλλογής προσώπων και η εύρεση προσώπων μέσα σε αυτήν δεν είναι δυνατή.



## 2. CompreFace



### Exadel CompreFace

Το CompreFace είναι ένα δωρεάν και ανοιχτού κώδικα project αναγνώρισης προσώπου στο GitHub. Ουσιαστικά, πρόκειται για μια εφαρμογή βασισμένη σε docker\* η οποία μπορεί να χρησιμοποιηθεί είτε ως αυτόνομος διακομιστής είτε να αναπτυχθεί στο cloud. Δεν είναι προαπαιτούμενες οι γνώσεις μηχανικής μάθησης για την ρύθμιση και την λειτουργία του.

Το σύστημα παρέχει REST API για αναγνώριση προσώπου, επιβεβαίωση προσώπου, ανίχνευση προσώπου, ανίχνευση ορόσημων, αναγνώρισης φύλου και ηλικίας. Διαθέτει επίσης ένα σύστημα διαχείρισης ρόλων, το οποίο επιτρέπει στον χρήστη να ελέγχει εύκολα ποιος έχει πρόσβαση στις Υπηρεσία Αναγνώρισης Προσώπου του.

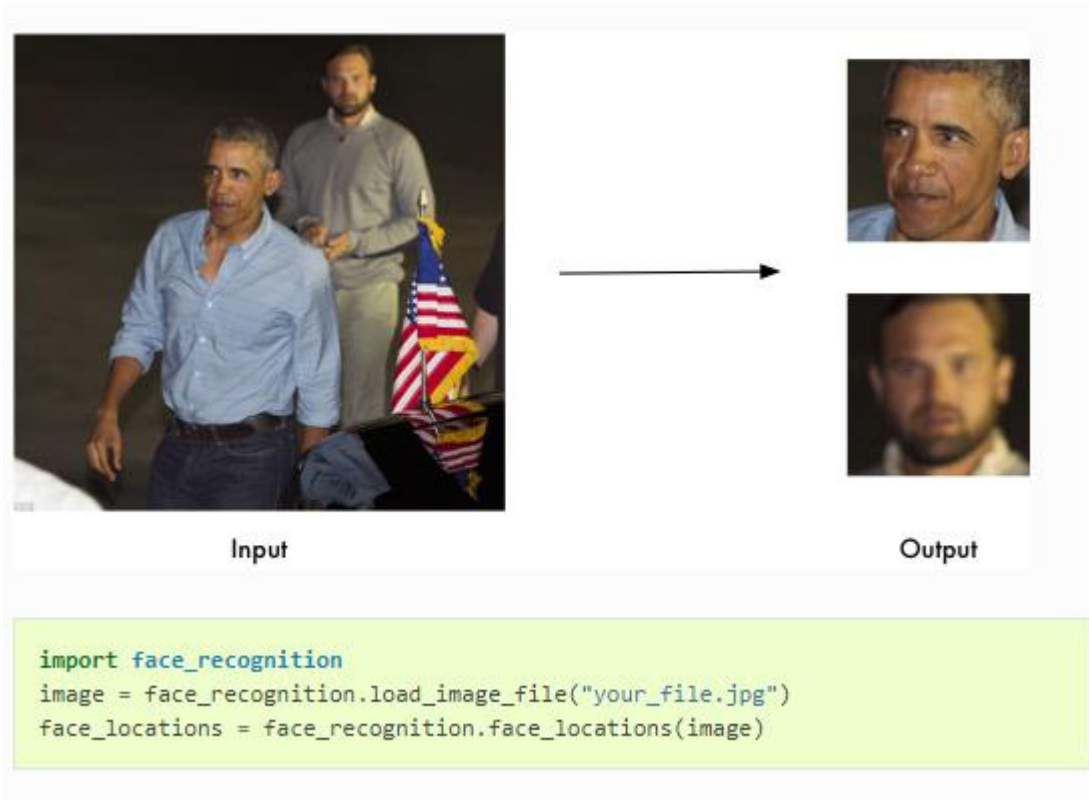
Διανέμεται σαν μια διάταξη βασισμένη σε docker\* και υποστηρίζει διάφορα μοντέλα τα οποία λειτουργούν είτε σε CPU είτε σε GPU.

Το CompreFace είναι βασισμένο σε μεθόδους τελευταίας τεχνολογίας και σε βιβλιοθήκες όπως FaceNet και InsightFace.

Έχει ποσοστό ακριβείας 99,83% κάτι το οποίο το καθιστά μια αρκετά καλή επιλογή, ανάμεσα στις δύο πιο δημοφιλείς μεθόδους FaceNet (με ποσοστό ακρίβειας 99,65%) και InsightFace (με ποσοστό ακρίβειας 99,86%).

*docker\*: (είναι μια τεχνολογία λειτουργικού συστήματος που επιτρέπει στις εφαρμογές να τοποθετούνται σαν στοιβαγμένα κουτιά)*

### 3. Face Recognition



Αναγνωρίζει και διαχειρίζεται τα πρόσωπα μέσω του Python API ή μέσω του δυαδικού εργαλείου γραμμής εντολών. Κατασκευάστηκε χρησιμοποιώντας την υπερσύγχρονη αναγνώριση προσώπου της dlib, δομημένο με βαθιά μάθηση. Το μοντέλο έχει ποσοστό ακρίβειας 99,38%. Επίσης, παρέχει ένα απλό εργαλείο γραμμής εντολών: `face_recognition` το οποίο επιτρέπει την αναγνώριση προσώπου σε έναν φάκελο εικόνων από την γραμμή εντολών.

Ορισμένες δυνατότητες που παρέχει είναι:

- Εύρεση των προσώπων στην εικόνα.
- Εύρεση και διαχείριση των χαρακτηριστικών του προσώπου στην εικόνα.
- Ταυτοποίηση των προσώπων στην εικόνα.

Μπορεί να πραγματοποιηθεί χρήση της βιβλιοθήκης αυτής σε συνδυασμό με άλλες βιβλιοθήκες Python, ώστε να γίνει αναγνώριση προσώπου σε πραγματικό χρόνο.

#### 4. InsightFace



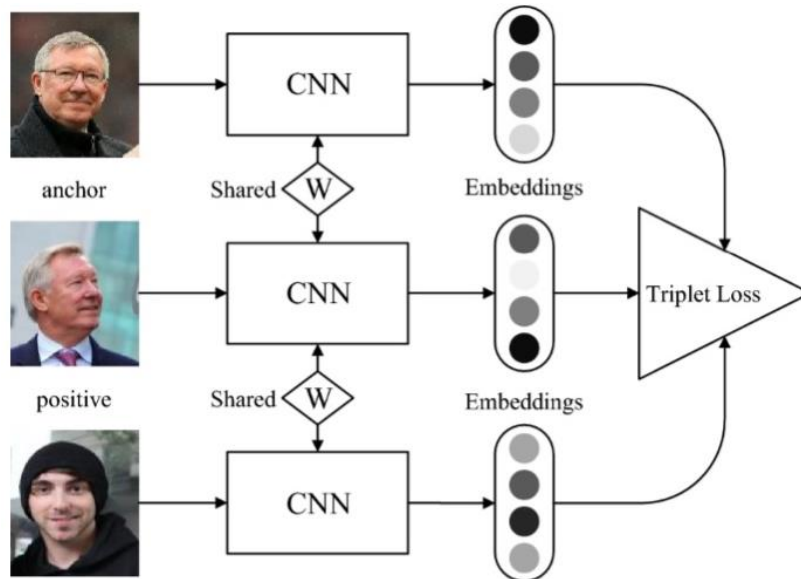
Πρόκειται για μια υπερσύγχρονη βιβλιοθήκη δυσδιάστατης και τρισδιάστατης βαθιάς ανάλυσης προσώπου, ενσωματωμένη σε γλώσσα προγραμματισμού Python. Το InsightFace εφαρμόζει αποτελεσματικά μια ποικιλία από τελευταίας γενιάς αλγορίθμους αναγνώρισης προσώπου (SubCenter-ArcFace), εντοπισμού προσώπου (RetinaFace) και ευθυγράμμισης προσώπου, οι οποίοι βελτιστοποιήθηκαν τόσο για εκπαίδευση όσο για ανάπτυξη. Τα ερευνητικά ινστιτούτα και οι βιομηχανικοί οργανισμοί μπορούν να επωφεληθούν από την χρήση της βιβλιοθήκης αυτής.

Κατέχει υψηλό ποσοστό ακρίβειας της τάξης του 99,89%.

Ο κώδικας του InsightFace κυκλοφορεί υπό την άδεια του MIT και δεν υπάρχει περιορισμός τόσο για ακαδημαϊκή όσο για εμπορική χρήση.



## 5. FaceNet



Δημιουργήθηκε από μια ομάδα ερευνητών της Google, είναι μια δημοφιλής ανοιχτού κώδικα βιβλιοθήκη σε γλώσσα προγραμματισμού Python. Ο κώδικας είναι εμπνευσμένος σε μεγάλο βαθμό από την υλοποίηση του OpenFace. Πρόκειται για μια αρκετά απλή βιβλιοθήκη για την αναγνώριση προσώπων από εικόνες.

Οι δυνατότητες που παρέχει είναι:

- Εντοπισμός προσώπου, γίνεται με χρήση του αλγόριθμου MTCNN.
- Ευθυγράμμιση προσώπου, γίνεται ευθυγράμμιση σύμφωνα με την γραμμή των ματιών.
- Κωδικοποίηση προσώπου, γίνεται εξαγωγή κωδικοποίησης από τα πρόσωπα με χρήση του FaceNet.
- Ταξινόμηση προσώπου, γίνεται ταξινόμηση του προσώπου με χρήση ευκλείδειας μετρικής μεταξύ των κωδικοποιήσεων προσώπου.

Η ακρίβεια αυτού του μοντέλου είναι αρκετά υψηλή με ποσοστό 99,65%.

Παρότι πρόκειται για μια μέθοδο σχετικά παλιά, νέοι ερευνητές την χρησιμοποιούν ακόμα. Πρόσφατο παράδειγμα είναι το project «Αναγνώριση Προσώπου σε μάσκα» (MaskTheFace\*), όπου γίνεται μετατροπή συλλογής δεδομένων προσώπου σε συλλογή δεδομένων προσώπου με μάσκα.

MaskTheFace\*: <https://github.com/ageelanwar/MaskTheFace>

## 6. InsightFace-REST

# SthPhoenix/InsightFace-REST



InsightFace REST API for easy deployment of face recognition services with TensorRT in Docker.

1

Contributor

3

Issues

218

Stars

42

Forks



Όπως και το ComprFace, πρόκειται για μια εφαρμογή βασισμένη σε docker\* που παρέχει ένα βολικό REST API. Το μεγαλύτερο πλεονέκτημα της είναι ότι οι προγραμματιστές κατάφεραν να επιταχύνουν την αναγνώριση προσώπου του InsightFace επί τρεις φορές. Το μειονέκτημα της εφαρμογής αυτής είναι ότι παρέχει μόνο ενσωματώσεις προσώπων (face embeddings), δεν δίνει το API για την αναγνώριση προσώπου, οπότε θα πρέπει ο χρήστης να παρέχει τον δικό του ταξινομητή. Την δεδομένη στιγμή το αποθετήριο δεν διαθέτει άδεια χρήσης, οπότε θα πρέπει να πάρει άδεια ο χρήστης από τον συγγραφέα για να μπορέσει να το χρησιμοποιήσει.

*docker\**: (είναι μια τεχνολογία λειτουργικού συστήματος που επιτρέπει στις εφαρμογές να τοποθετούνται σαν στοιβαγμένα κουτιά)

## Αναφορά σε άλλες εφαρμογές

Αξίζει να γίνει αναφορά και σε άλλες δωρεάν γνώστες εφαρμογές αναγνώρισης προσώπου, όπως:

- OpenBR (open-source biometric recognition, C++ API, supports Windows, Linux and Mac)
- OpenFaceTracker (open-source API, supports Windows based systems)
- OpenEBTS (open-source API [OpenM1 API, OpenEBTS API], supports Windows based systems, Linux and Android)
- Bionable Tech iFace (open-source API, web-based)
- Bionable Tech vFace (web-based)
- Flandmark (open-source C Library, supports Windows and Linux)
- Face++ free version (popular API that also has an offline SDK for iOS and Android, offline SDK doesn't provide face recognition)

Επίσης, ορισμένες αρκετά δημοφιλείς εφαρμογές αναγνώρισης προσώπου επί πληρωμή, είναι οι εξής:

- Amazon Rekognition
- Deep Vision AI
- FaceFirst
- Face++
- FaceX
- Kairos
- Microsoft Azure Cognitive Services Face API

## 2. Η πλατφόρμα OpenCV σε γλώσσα προγραμματισμού Python

### 2.1. OpenCV

Η OpenCV είναι μια τεράστια βιβλιοθήκη ανοιχτού κώδικα, χρησιμοποιείται σε εφαρμογές υπολογιστικής όρασης, μηχανικής μάθησης και επεξεργασίας εικόνας[17]. Επίσης, αξίζει να σημειωθεί ότι παίζει ιδιαίτερα σημαντικό ρόλο σε εφαρμογές πραγματικού χρόνου.

Προτού εμβαθύνουμε περαιτέρω στην σημασία της βιβλιοθήκης αυτής, θα γίνει αναφορά στην έννοια των όρων «υπολογιστική όραση» (computer vision) και «επεξεργασία εικόνας» (image processing).

#### **Υπολογιστική Όραση (Computer Vision)**

Η υπολογιστική όραση αναφέρεται σε μια διαδικασία η οποία επιτρέπει σε κάποιον να κατανοήσει τις εικόνες και τα βίντεο, συγκεκριμένα πώς αποθηκεύονται, πώς μπορούν να χειραγωγηθούν και να ανακτηθούν δεδομένα από αυτά. Η υπολογιστική όραση είναι το θεμέλιο της τεχνητής νοημοσύνης. Την σήμερα ημέρα, παίζει σημαντικό ρόλο σε εφαρμογές όπως: αυτοοδηγούμενα οχήματα, ρομποτική, εφαρμογές διόρθωσης φωτογραφιών και πολλά άλλα.

#### **Επεξεργασία Εικόνας (Image Processing)**

Η επεξεργασία εικόνας αναφέρεται στην ανάλυση και στην χειραγώγηση μιας ψηφιοποιημένης εικόνας, συγκεκριμένα για την βελτίωση της ποιότητας της. Περιλαμβάνει διαδικασίες οι οποίες εκτελούνται σε εικόνες, προκειμένου να ληφθούν οι βελτιστοποιημένες εκδόσεις τους ή να συλλεχθούν/ εξαχθούν χρήσιμες πληροφορίες από αυτές. Τέτοιες διαδικασίες περιλαμβάνουν τα βήματα: εισαγωγής της εικόνας, ανάλυσης και χειραγώγησης της και λήψη της εξόδου, που συνήθως είναι μια τροποποιημένη εικόνα ή μια αναφορά βασισμένη στην ανάλυση της εικόνας. Η επεξεργασία εικόνας βοηθάει σε εργασίες όπως είναι η ανάγνωση (reading) και η εγγραφή (writing) εικόνων, ανίχνευση/εντοπισμός προσώπων και των χαρακτηριστικών τους, ανίχνευση σχημάτων όπως τετράγωνα, κύκλοι κ.α. (παράδειγμα: εντοπισμός νομισμάτων στις εικόνες), αναγνώριση κειμένου στις εικόνες (παράδειγμα: ανάγνωση πινακίδων οχημάτων), αλλαγή ποιότητας της εικόνας και των χρωμάτων/ φίλτρων (παράδειγμα: Instagram) κ.α. και ανάπτυξη λειτουργιών ή εφαρμογών Επαυξημένης Πραγματικότητας (Augmented Reality).

Η ονομασία της βιβλιοθήκης OpenCV προέρχεται από το Open Source Computer Vision. Εν συντομία, είναι μια βιβλιοθήκη η οποία χρησιμοποιείται για την επεξεργασία εικόνας. Όπως αναφέραμε παραπάνω, πρόκειται για μια τεράστια βιβλιοθήκη ανοιχτού κώδικα η οποία χρησιμοποιείται για εφαρμογές υπολογιστικής όρασης, σε τομείς που κάνουν χρήση αλγορίθμων τεχνητής νοημοσύνης ή μηχανικής μάθησης και για την ολοκλήρωση εργασιών που απαιτούν επεξεργασία εικόνας. Με αποτέλεσμα να παίζει σημαντικό ρόλο στα σημερινά συστήματα που αφορούν διαδικασίες πραγματικού χρόνου. Με την βοήθεια της είναι δυνατή η επεξεργασία εικόνας ή βίντεο με σκοπό την αναγνώριση αντικειμένων, προσώπων, ακόμη και του ανθρώπινου γραφικού χαρακτήρα.

Αρχικά αναπτύχθηκε από την Intel, έπειτα υποστηρίχθηκε από άλλες εταιρίες, στην πορεία όμως επέστρεψε πάλι στην ιδιοκτησία της Intel. Αρχικά, κυκλοφόρησε με άδεια χρήσης BSD (Berkeley Software Distribution/ Berkeley Standard Distribution), παρέχεται δωρεάν τόσο για ακαδημαϊκή όσο για εμπορική χρήση σύμφωνα με την άδεια χρήσης ανοιχτού κώδικα APACHE 2. Διαθέτει διεπαφές σε γλώσσες προγραμματισμού C, C++, Python, Java και MATLAB, επίσης το API για τις διεπαφές αυτές μπορεί να βρεθεί στο διαδικτυακό εγχειρίδιο της βιβλιοθήκης. Υποστηρίζει λειτουργικά συστήματα Windows, Linux και Mac OS για υπολογιστές, καθώς επίσης iOS, Android για κινητά και tablets. Ο αρχικός στόχος της βιβλιοθήκης αυτής ήταν η δημιουργία εφαρμογών πραγματικού χρόνου με σκοπό την υπολογιστική απόδοση. Πλέον, προσφέρει επιτάχυνση της μονάδας επεξεργασίας γραφικών για διαδικασίες πραγματικού χρόνου. Με την ενσωμάτωση άλλων βιβλιοθηκών όπως είναι η Numpy (βιβλιοθήκη της γλώσσας προγραμματισμού Python), η Python μπορεί να επεξεργαστεί την δομή του πίνακα της OpenCV για ανάλυση. Η αναγνώριση μοτίβων εικόνων και διαφόρων χαρακτηριστικών της, απαιτεί τη χρήση διανυσματικού χώρου και την υλοποίηση μαθηματικών πράξεων σε αυτά τα χαρακτηριστικά.

Ορισμένα παραδείγματα εφαρμογών της OpenCV, είναι τα εξής:

- Εντοπισμός προσώπου, αναγνώριση προσώπου, συστήματα αναγνώρισης προσώπου
- Αναγνώριση χειρονομιών
- Αλληλεπίδραση ανθρώπου-υπολογιστή (γνωστό ως HCI)
- Κινητή ρομποτική
- Κατανόηση κίνησης
- Εντοπισμός αντικειμένων
- Διαχωρισμός και αναγνώριση εικόνας
- Παρακολούθηση κίνησης
- Επαυξημένη Πραγματικότητα (γνωστό ως AR)

Ορισμένα Πλεονεκτήματα χρήσης της βιβλιοθήκης OpenCV, είναι τα εξής:

- Είναι εύκολη στην χρήση και στην εκμάθηση.
- Υπάρχει πληθώρα παραδειγμάτων/σεμιναρίων στο διαδίκτυο.
- Υπάρχει συμβατότητα με ευρέως διαδεδομένες γλώσσες προγραμματισμού.
- Διανέμεται δωρεάν.

Συγκεκριμένα, επιλέχθηκε η βιβλιοθήκη αυτή λόγο ενός βασικού αλγορίθμου που χρησιμοποιεί για την αναγνώριση προσώπων, ο οποίος βασίζεται στην μέθοδο «Viola and Jones»[18]–[22].

Ο αλγόριθμος αυτός πήρε την ονομασία του από τους δύο ερευνητές Paul Viola και Michael Jones το 2001 με τη δημοσίευση της εργασίας τους με τίτλο «Rapid Object Detection using a Boosted Cascade of Simple Features»[23]. Αν και πρόκειται για ένα ξεπερασμένο framework, είναι αρκετά ισχυρός και η εφαρμογή του για τον εντοπισμό προσώπου σε πραγματικό χρόνο έχει αποδειχθεί υψίστης σημασίας. Ο αλγόριθμος είναι αρκετά αργός κατά την διαδικασία της εκπαίδευσης αλλά μπορεί να εντοπίσει πρόσωπα σε πραγματικό χρόνο με εκπληκτική ταχύτητα.

Καταρχάς, ο αλγόριθμος αυτός λειτουργεί με ασπρόμαυρες εικόνες, οπότε απαιτείται η μετατροπή της τρέχουσας εικόνας από RGB σε GRAYSCALE. Όταν του δοθεί εικόνα (ασπρόμαυρη), τότε αυτός εξετάζει μικρότερες υπό-περιοχές της εικόνας αυτής και προσπαθεί να βρει ένα πρόσωπο αναζητώντας συγκεκριμένα χαρακτηριστικά σε κάθε υπό-περιοχή. Απαιτείται ο έλεγχος πολλών διαφορετικών θέσεων και κλίμακας μεγεθών, καθώς μια εικόνα μπορεί να περιέχει περισσότερα

από ένα πρόσωπα διαφόρων μεγεθών (απόσταση από την κάμερα). Ο αλγόριθμος αυτός κάνει χρήση των χαρακτηριστικών τύπου Haar για τον εντοπισμό προσώπων.

Αποτελείται από τέσσερα βασικά βήματα, τα οποία είναι τα εξής:

1. Επιλογή χαρακτηριστικών τύπου Haar
2. Δημιουργία εικόνας ακεραίων αριθμών (integral image)
3. Εκτέλεση εκπαίδευσης «AdaBoost»
4. Δημιουργία καταρράκτη ταξινομητών

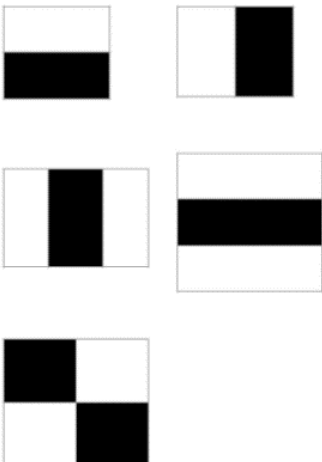
## 2.2. Χαρακτηριστικά τύπου Haar

Οι ερευνητές Paul Viola και Michael Jones[23], [24] βασίστηκαν στην έρευνα του Alfred Haar (κουματίδια Haar) και ανέπτυξαν τα λεγόμενα «χαρακτηριστικά τύπου Haar»[25]–[27].

Πρόκειται για χαρακτηριστικά ψηφιακής εικόνας, τα οποία χρησιμοποιούνται για τον εντοπισμό αντικειμένων. Υπάρχουν κάποιες καθολικές ιδιότητες οι οποίες εντοπίζονται στα πρόσωπα όλων των ανθρώπων, ορισμένα παραδείγματα αυτών είναι: η περιοχή των ματιών είναι πιο σκούρα από τα γειτονικά pixels, η περιοχή της μύτης είναι πιο φωτεινή από την περιοχή των ματιών κ.α.

Ο πιο απλός τρόπος για τον εντοπισμό της πιο σκουρόχρωμης και της πιο ανοιχτόχρωμης περιοχής είναι να γίνει άθροιση των τιμών των pixels για την κάθε περιοχή και έπειτα να γίνει η σύγκριση των τιμών που προήλθαν από τα αθροίσματα. Το άθροισμα των τιμών των pixels της σκουρόχρωμης περιοχής θα είναι μικρότερο από το άθροισμα των τιμών των pixels της ανοιχτόχρωμης περιοχής. Ενημερωτικά, στο RGB model το νούμερο #000000 αντιπροσωπεύει το μαύρο χρώμα και το #FFFFFF το λευκό χρώμα. Για παράδειγμα, αν μια περιοχή είναι πιο ανοιχτόχρωμη από μια άλλη, αυτό μπορεί να αντιπροσωπεύει την άκρη του φρυδιού. Ένα άλλο παράδειγμα είναι, αν μια μεσαία περιοχή είναι πιο φωτεινή από τα γειτονικά pixels, τότε αυτό μπορεί να αντιστοιχεί στην περιοχή της μύτης. Όλα αυτά μπορούν να γίνουν με την χρήση των «χαρακτηριστικών τύπου Haar» και με την βοήθεια τους μπορούν να ερμηνευτούν οι διάφορες περιοχές του προσώπου.

Υπάρχει μια πληθώρα από διαφορετικούς τύπους χαρακτηριστικών τύπου Haar[28], μέσα από αυτούς διακρίνονται τρεις βασικοί τύποι:

- |  |  |   |
|--|--|---|
| <ol style="list-style-type: none"> <li>1. Χαρακτηριστικά Ακμών/Άκρων</li> <li>2. Χαρακτηριστικά Γραμμής</li> <li>3. Χαρακτηριστικά Τεσσάρων Όψεων</li> </ol> |  | <p>1. Edge Features</p> <p>2. Line Features</p> <p>3. Four rectangle Features</p> |
|--|--|---|



Από την ονομασία τους είναι προφανές ότι τα χαρακτηριστικά ακμών χρησιμοποιούνται για τον εντοπισμό ακμών, τα χαρακτηριστικά γραμμής χρησιμοποιούνται για τον εντοπισμό γραμμών. Τέλος τα χαρακτηριστικά Τεσσάρων όψεων χρησιμοποιούνται για τον εντοπισμό διαγώνιων χαρακτηριστικών.

Η τιμή των χαρακτηριστικών υπολογίζεται από έναν μοναδικό αριθμό, ο οποίος προκύπτει από την ακόλουθη πράξη: το άθροισμα των τιμών των pixel στην σκουρόχρωμη περιοχή μείον το άθροισμα των τιμών των pixel στην ανοιχτόχρωμη περιοχή. Στην περίπτωση μιας απλής επιφάνειας στην οποία όλα τα pixel έχουν την ίδια τιμή, τότε η τιμή του χαρακτηριστικού είναι μηδέν καθώς δεν περιέχει καμία χρήσιμη πληροφορία. Ένα χαρακτηριστικό τύπου Haar έχει μεγάλη τιμή όταν οι σκουρόχρωμες και οι ανοιχτόχρωμες περιοχές είναι διαφορετικές. Χρησιμοποιώντας την τιμή αυτή λαμβάνεται έγκυρη πληροφορία από την εικόνα.

Υπάρχουν κάποια γνωστά χαρακτηριστικά τα οποία είναι ιδιαίτερα αποτελεσματικά για τον εντοπισμό προσώπων. Για παράδειγμα, με την εφαρμογή ενός συγκεκριμένου χαρακτηριστικού τύπου Haar στην περιοχή της μύτης, υπάρχει υψηλή απόκριση. Με παρόμοιο τρόπο, αν γίνει συνδυασμός πολλών τέτοιων χαρακτηριστικών τύπου Haar τότε μπορεί να γίνει κατανοητό αν μια συγκεκριμένη περιοχή της εικόνας περιέχει πρόσωπο.

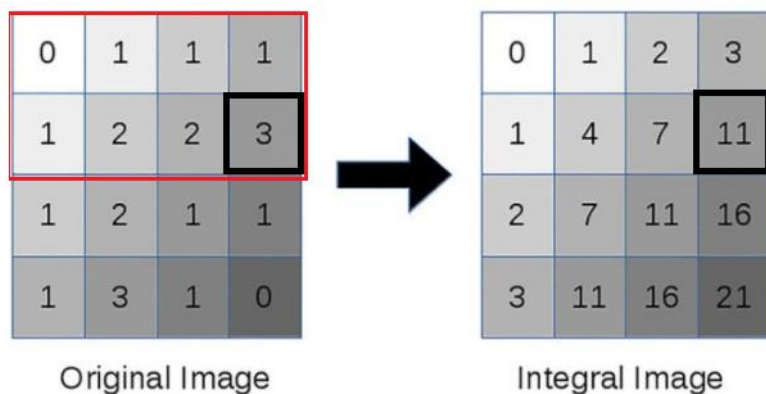
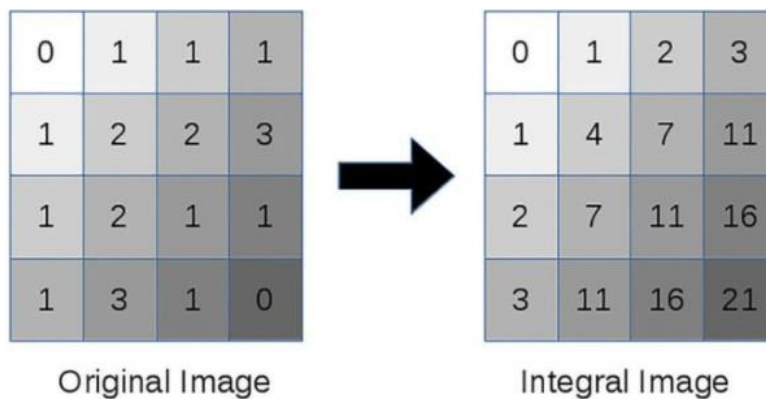
### 2.3. Εικόνες Ακεραίων Αριθμών (Integral Images)

Όπως αναφέρθηκε προηγουμένως, για να βρεθεί η τιμή κάθε χαρακτηριστικού πρέπει να γίνει υπολογισμός όλων των τιμών των pixel που βρίσκονται μέσα στο συγκεκριμένο χαρακτηριστικό. Οι υπολογισμοί αυτοί είναι εντατικοί όταν πρόκειται για μεγάλα χαρακτηριστικά, τα οποία έχουν σαφώς μεγαλύτερο αριθμό pixel.

Η εικόνα ακεραίων αριθμών είναι ιδιαίτερα σημαντική γιατί δίνει την δυνατότητα να εκτελεστούν αυτοί οι εντατικοί υπολογισμοί σε γρήγορο χρονικό διάστημα, με σκοπό να γίνει κατανοητό αν ένα χαρακτηριστικό πληροί τα κριτήρια.

Αποκαλείται και «πίνακας αθροιστικής περιοχής», έχει διπλή σημασία τόσο δομή δεδομένων όσο και αλγόριθμου, ο οποίος χρησιμοποιείται για την απόκτηση της δομής δεδομένων. Χρησιμοποιείται ως ένας γρήγορος τρόπος για τον υπολογισμό των τιμών των pixel που περιέχονται σε μια εικόνα ή σε μια περιοχή της εικόνας.

Η τιμή του κάθε σημείου είναι το άθροισμα όλων των τιμών των pixel με φορά προς τα πάνω και προς τα αριστερά, συμπεριλαμβανομένου του επιθυμητού pixel.



Γίνεται εξοικονόμηση χρόνου, καθώς για τον υπολογισμό του αθροίσματος όλων των τιμών των pixel σε ένα τετράγωνο, αρκεί να γίνει υπολογισμός των τεσσάρων ακμών του τετραγώνου.

0	1	1	1	0	1	2	3
1	2	2	3	1	4	7	11
1	2	1	1	2	7	11	16
1	3	1	0	3	11	16	21

Στην παραπάνω εικόνα, στον πρώτο πίνακα (original image) αθροίζοντας όλες τις τιμές των pixel που περιέχονται μέσα στο γαλάζιο πλαίσιο, το αποτέλεσμα της πράξης είναι η τιμή 8. Για τον υπολογισμό του αθροίσματος των αντίστοιχων pixel στον δεύτερο πίνακα της εικόνας ακεραίων αριθμών πρέπει να βρεθούν οι γωνίες του τετραγώνου και έπειτα να προστεθούν οι διαγώνιες κορυφές με πράσινο χρώμα και να αφαιρεθούν οι διαγώνιες κορυφές με κόκκινο χρώμα. Η πράξη είναι η εξής:

$$1+21-11-3=8$$

Παρατηρείται ότι λαμβάνεται η ίδια απάντηση (τιμή 8) αλλά στην περίπτωση αυτή ο υπολογισμός γίνεται με τέσσερις μόνο αριθμούς. Ανεξάρτητα από το πλήθος των pixel που εμπεριέχονται μέσα στο πλαίσιο, αρκεί να υπολογιστούν οι τέσσερις κορυφές.

Με την μέθοδο αυτή, ο υπολογισμός της τιμής οποιουδήποτε χαρακτηριστικού τύπου Haar γίνεται εύκολα υπολογίζοντας τη διαφορά μεταξύ των αθροισμάτων των τιμών των pixel δυο ορθογωνίων.

## 2.4. Αλγόριθμος AdaBoost

Έπειτα, γίνεται χρήση ενός αλγορίθμου μηχανικής μάθησης, του αλγορίθμου AdaBoost[26], [27]. Σε ένα παράθυρο ανιχνευτή διαστάσεων 24x24 ο αριθμός των χαρακτηριστικών που εμπεριέχονται ξεπερνούν τα 160.000, αλλά μόνο ορισμένα από αυτά είναι απαραίτητα για την αναγνώριση προσώπου. Για τον λόγο αυτό γίνεται χρήση του αλγορίθμου AdaBoost, ώστε να εντοπιστούν τα καλύτερα χαρακτηριστικά από τα 160.000.

Στον αλγόριθμο Viola and Jones κάθε χαρακτηριστικό τύπου Haar αντιπροσωπεύει έναν «αδύναμο» ταξινομητή. Ενώ, ο αλγόριθμος AdaBoost ελέγχει την απόδοση όλων των ταξινομητών που παρέχονται από τον χρήστη, έτσι ώστε να αποφασίσει τον τύπο και το μέγεθος των χαρακτηριστικών τα οποία καταλήγουν στον τελικό ταξινομητή.

Αρχικά γίνεται αξιολόγηση του ταξινομητή σε όλες τις υπό-περιοχές όλων των εικόνων που χρησιμοποιούνται για την εκπαίδευση. Από όλες τις υπό-περιοχές ορισμένες θα παράγουν ισχυρή απόκριση στον ταξινομητή και αυτές θα ταξινομηθούν ως «θετικά», αυτό σημαίνει ότι ο ταξινομητής πιστεύει ότι περιέχουν πρόσωπο. Στην περίπτωση που οι υπό-περιοχές δεν παράγουν ισχυρή απόκριση ο ταξινομητής θεωρεί ότι δεν περιέχουν πρόσωπο και ταξινομούνται ως «αρνητικά».

Στους ταξινομητές που είχαν υψηλή απόδοση δίνεται «υψηλότερη σημασία» ή «βάρη». Το τελικό αποτέλεσμα είναι ένας ισχυρός ταξινομητής, ο οποίος περιέχει τους αποδοτικότερους «αδύναμους» ταξινομητές.

Στην ουσία, όταν γίνεται εκπαίδευση του αλγορίθμου AdaBoost για την αναγνώριση των σημαντικότερων χαρακτηριστικών, γίνεται τροφοδότηση πληροφοριών με την μορφή δεδομένων εκπαίδευσης, ακολούθως γίνεται εκπαίδευση του αλγορίθμου ώστε να προβλέπει από της πληροφορίες αυτές. Με τρόπο αυτό ο αλγόριθμος θέτει ένα ελάχιστο όριο έτσι ώστε να καθορίσει εάν κάτι θα ταξινομηθεί ως σημαντικό χαρακτηριστικό ή μη-σημαντικό χαρακτηριστικό.

## 2.5. Καταρράκτης Ταξινομητών

Ακόμη και αν ο αλγόριθμος AdaBoost επιλέξει τα καλύτερα χαρακτηριστικά (έστω τα 3.000) από τα συνολικά 160.000, δεν παύει να είναι μια χρονοβόρα διαδικασία για τον υπολογισμό αυτών των χαρακτηριστικών για κάθε περιοχή. Όπως αναφέραμε παραπάνω, ένα παράθυρο ανιχνευτή διαστάσεων 24x24, σκανάρει την εικόνα εισόδου με σκοπό την εύρεση των περιοχών που περιέχουν πρόσωπο. Ο καταρράκτης είναι υπεύθυνος για την γρήγορη απόρριψη περιοχών που δεν περιέχουν πρόσωπο, έτσι ώστε να μην χάνεται σημαντικός χρόνος και να μην γίνονται άσκοποι υπολογισμοί. Με τον τρόπο αυτό εξασφαλίζεται η απαραίτητη ταχύτητα για τον εντοπισμό προσώπου σε πραγματικό χρόνο.

Στην ουσία δημιουργείται ένα σύστημα καταρράκτη, του οποίου η διαδικασία αναγνώρισης προσώπων χωρίζεται σε πολλά επίπεδα. Στο πρώτο επίπεδο, υπάρχει ένας ταξινομητής ο οποίος αποτελείται από τα καλύτερα χαρακτηριστικά, στην ουσία η υπό-περιοχή κληρονομεί τα καλύτερα χαρακτηριστικά (π.χ. χαρακτηριστικά που προσδιορίζουν την περιοχή της μύτης ή χαρακτηριστικά που προσδιορίζουν την περιοχή των ματιών). Στα ακόλουθα επίπεδα, υπάρχουν όλα τα υπόλοιπα χαρακτηριστικά.

Όταν μια υπό-περιοχή της εικόνας εισάγεται στον καταρράκτη, αξιολογείται από το πρώτο επίπεδο. Στο επίπεδο αυτό, εάν αξιολογηθεί η υπό-περιοχή ως «θετική», αυτό σημαίνει ότι πιστεύει ότι περιέχει πρόσωπο, και η έξοδος του επιπέδου χαρακτηρίζεται ως «ίσως». Όταν μια υπό-περιοχή χαρακτηριστεί ως «ίσως», προχωράει στο επόμενο επίπεδο και η διαδικασία συνεχίζεται έως ότου φτάσει στο τελευταίο επίπεδο.

Στην περίπτωση που όλοι οι ταξινομητές «αποδεχθούν» την εικόνα, ταξινομείται ως πρόσωπο και ενημερώνεται ο χρήστης ότι υπάρχει ανίχνευση προσώπου.

Όλη αυτή η διαδικασία βοηθάει στην αύξηση της ταχύτητας των διεργασιών, καθώς αν στο πρώτο επίπεδο η υπό-περιοχή αξιολογηθεί ως «αρνητική», τότε η εικόνα απορρίπτεται άμεσα μιας και δεν περιέχει πρόσωπο. Επίσης αν, στο πρώτο επίπεδο η υπό-περιοχή αξιολογηθεί ως «θετική» αλλά αποτύχει στο δεύτερο επίπεδο, απορρίπτεται. Στην ουσία, η εικόνα μπορεί να απορριφθεί σε οποιοδήποτε επίπεδο του ταξινομητή.

### 3. Παρουσίαση Project

Για την πραγματοποίηση Αναγνώρισης Προσώπου με γλώσσα προγραμματισμού Python, απαιτείται η εγκατάσταση των παρακάτω βιβλιοθηκών:

- OpenCV
- Face\_recognition

Η εγκατάσταση της βιβλιοθήκης OpenCV γίνεται με την βοήθεια της εντολής pip, όπως φαίνεται παρακάτω:

```
$ pip install opencv-python
```

Αυτή η εντολή θα ξεκινήσει τη λήψη και την εγκατάσταση πακέτων που σχετίζονται με τη βιβλιοθήκη OpenCV[29]. Μόλις ολοκληρωθούν οι διαδικασίες, θα εμφανιστεί το μήνυμα επιτυχούς εγκατάστασης.

Η βιβλιοθήκη Dlib περιέχει την υλοποίηση της Βαθιάς Μάθησης, η οποία χρησιμοποιείται για την κατασκευή των ενσωματώσεων προσώπου που χρησιμοποιούνται για την διαδικασία αναγνώρισης.

Η βιβλιοθήκη face\_recognition συμπεριλαμβάνει τη λειτουργία αναγνώρισης προσώπου της βιβλιοθήκης Dlib διευκολύνοντας τις διεργασίες.

Η εγκατάσταση της βιβλιοθήκης face\_recognition γίνεται με την βοήθεια της εντολής pip, όπως φαίνεται παρακάτω:

```
$ pip install face_recognition
```

Παρακάτω γίνεται παρουσίαση του dataset των προσώπων για την διαδικασία αναγνώρισης προσώπου από το video που επιλέχθηκε ως είσοδος στον κώδικα. Η δομή του dataset έχει ως εξής:

```
$ tree images -filelimit 9
```

```
|— images  
|   |— Benicio Del Toro [10 entries]  
|   |— Joel Edgerton [10 entries]  
|   |— Mark Ruffalo [10 entries]  
|   |— Matthew Belloni [10 entries]  
|   |— Michael Caine [10 entries]  
|   |— Samuel L. Jackson [10 entries]  
|   |— Stephen Galloway [10 entries ]  
|   └─ Will Smith [10 entries]
```

Για την αναγνώριση προσώπου λαμβάνεται δείγμα 10 εικόνων για το κάθε άτομο που περιέχεται μέσα στο video.

Ο φάκελος «images» περιέχει εικόνες των προσώπων οχτώ ατόμων, με σκοπό την αναγνώριση των ατόμων κατά την διάρκεια του video.

Η δομή του project έχει ως εξής:

```
|— main_recognition.py  
|— images  
|   |— Benicio Del Toro [10 entries]  
|   |— Joel Edgerton [10 entries]  
|   |— Mark Ruffalo [10 entries]  
|   |— Matthew Belloni [10 entries]  
|   |— Michael Caine [10 entries]  
|   |— Samuel L. Jackson [10 entries]  
|   |— Stephen Galloway [10 entries]  
|   └─ Will Smith [10 entries]  
└─ video.mp4
```

video.mp4: Το αρχείο βίντεο που χρησιμοποιήθηκε, το οποίο δεν έχει υποβληθεί ακόμα στο σύστημα αναγνώρισης προσώπου.

images: Ο φάκελος με τις εικόνες των προσώπων των ατόμων.

main\_recognition.py: Το πρόγραμμα το οποίο θα εκτελεστεί για την αναγνώριση των προσώπων.

Για την εκτέλεση του παρακάτω κώδικα, χρησιμοποιήθηκε video συνέντευξης του ηθοποιού Will Smith το οποίο λήφθηκε από την πλατφόρμα YouTube και στην πορεία επεξεργάστηκε ώστε να μικρύνει η διάρκεια του στα 38sec (στην πραγματικότητα η διάρκεια της συνέντευξης είναι 33sec, τα 4sec είναι από το πρόγραμμα επεξεργασίας που χρησιμοποιήθηκε). Η επεξεργασία του video έγινε έτσι ώστε να μπορέσει να τρέξει ο κώδικας σε σύντομο χρονικό διάστημα.

Για καλύτερες επιδόσεις, καθώς απαιτείται μεγάλη υπολογιστική ισχύ της GPU, ο κώδικας έτρεξε στο περιβάλλον «Colab» του Google-Chrome.

Το «Colab» είναι ένα προϊόν της Google (συγκεκριμένα της Google Research) και επιτρέπει στους χρήστες να γράφουν και να εκτελούν κώδικα σε γλώσσα προγραμματισμού Python στο πρόγραμμα περιήγησης τους (browser) χωρίς να απαιτείται διαμόρφωση (configuration), με δωρεάν πρόσβαση σε GPUs και κάνοντας εύκολα κοινή χρήση με άλλους χρήστες.

Τα Colab notebooks επιτρέπουν στους χρήστες τον συνδυασμό εκτελέσιμου κώδικα και εμπλουτισμένου κειμένου σε ένα μόνο αρχείο με εικόνες, HTML, LaTeX κ.α. Όταν ο χρήστης δημιουργεί τα δικά του Colab notebooks αυτά αποθηκεύονται στον Google λογαριασμό του στο Google Drive. Πράγμα το οποίο καθιστά εύκολη την κοινοποίηση των project με συναδέλφους ή φίλους, επιτρέποντας τον σχολιασμό σε αυτά καθώς και την επεξεργασία τους.

Τα Colab notebooks είναι Jupyter notebooks που φιλοξενούνται από την Colab.

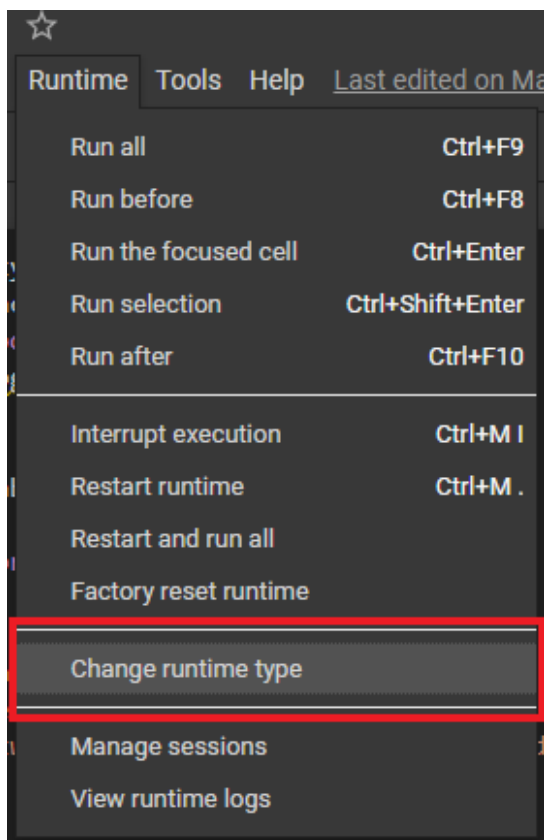
Συγκεκριμένα για την Μηχανική Μάθηση (Machine Learning): Με το Colab ο χρήστης μπορεί να εισάγει ένα σύνολο δεδομένων εικόνας (image dataset), να εκπαιδεύσει έναν ταξινομητή εικόνας (image classifier) και να αξιολογήσει το μοντέλο σε λίγες γραμμές κώδικα. Τα Colab notebooks εκτελούν κώδικα στους cloud servers της Google, αυτό σημαίνει ότι ο χρήστης μπορεί να αξιοποιήσει τις παροχές hardware της Google, συμπεριλαμβανομένων των GPUs και των TPUs, ασχέτως από την υπολογιστική δύναμη του υπολογιστή που διαθέτει. Η μοναδική προϋπόθεση είναι να διαθέτει ένα πρόγραμμα περιήγησης (browser).



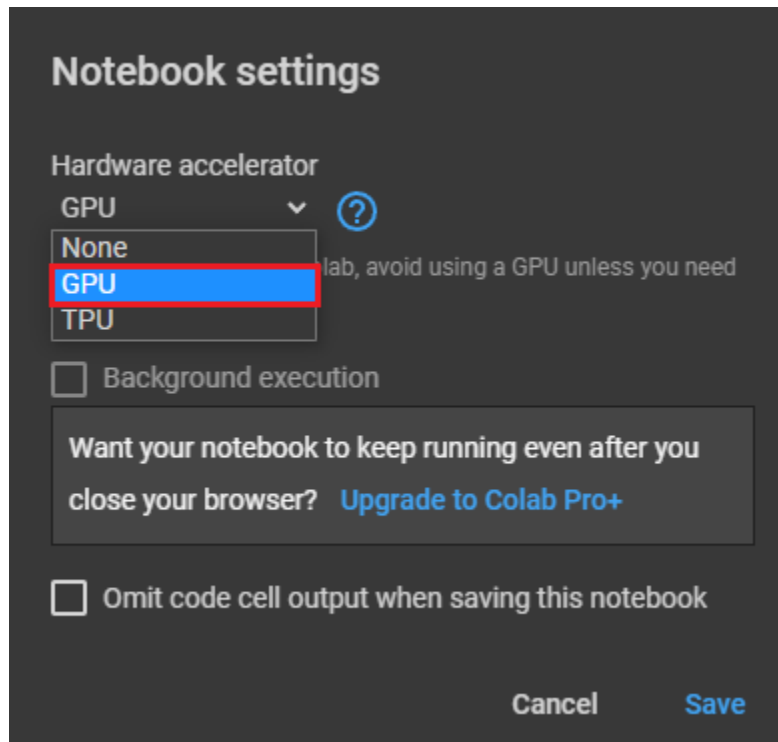
Το Colab χρησιμοποιείται ευρέως στην κοινότητα της Μηχανικής Μάθησης, συγκεκριμένα σε εφαρμογές όπως:

- Εισαγωγή στο TensorFlow
- Ανάπτυξη και Εκπαίδευση Νευρωνικών Δικτύων
- Πειραματισμός με TPUs κ.α.

Πριν γίνει η εκτέλεση του κώδικα επιλέγουμε από το taskbar -> Runtime -> Change runtime type



Έπειτα, στο αναδυόμενο παράθυρο επιλέγουμε: Hardware accelerator -> GPU



Αρχικά δίνονται οι παρακάτω εντολές έτσι ώστε να γίνει αποσυμπίεση του φακέλου που περιέχει τις εικόνες και εγκατάσταση της βιβλιοθήκης face\_recognition:

```
!unzip images.zip
!pip3 install face_recognition
```

Έπειτα γίνεται εισαγωγή των απαραίτητων πακέτων:

```
from pathlib import Path
import face_recognition
import cv2
import csv
from google.colab.patches import cv2_imshow
import json
```

```
from typing import Union, List
```

**Pathlib:** Η ενότητα «pathlib» της Python καθιστά πολύ εύκολη και αποτελεσματική την αντιμετώπιση διαδρομών αρχείων. Η ενότητα διαδρομής «os» μπορεί επίσης να χρησιμοποιηθεί για τον χειρισμό λειτουργιών ονόματος διαδρομής. Η διαφορά είναι ότι η ενότητα διαδρομής «os» δημιουργεί συμβολοσειρές που αντιπροσωπεύουν διαδρομές αρχείων, ενώ η «pathlib» δημιουργεί ένα αντικείμενο διαδρομής.

**Typing:** Οι υποδείξεις τύπου (type hints) βοηθούν στη δημιουργία και στη διατήρηση μιας ευανάγνωστης αρχιτεκτονικής.

**Csv:** Η λειτουργική μονάδα «csv» εφαρμόζει τάξεις (classes) για την ανάγνωση και την εγγραφή δεδομένων πίνακα σε μορφή CSV. Επιτρέπει στους προγραμματιστές να δίνουν εντολές εγγραφής των δεδομένων στην μορφή που ορίζει το Excel, καθώς επίσης και εντολές ανάγνωσης δεδομένων από αρχείο το οποίο δημιουργήθηκε στο περιβάλλον Excel, χωρίς να γνωρίζουν τις ακριβείς λεπτομέρειες της μορφής CSV που χρησιμοποιείται από το Excel.

Στην συνέχεια, έχουμε την εξής μέθοδο:

```
def recognition_faces_image(images_path: Path) -> dict:
    """Recognize faces in a single image based on dataset.
    Initialize two lists will contain the face encodings and corresponding names for each person in
    the dataset

    Args:
        images_path (Path): Path of Images (Dataset)

    Returns:
        dict: encodings and names
    """
    all_people_images = [image for folder in images_path.iterdir()
                          for image in folder.iterdir()]
```

```

known_encodings = []
known_names = []

for image in all_people_images:

    img = cv2.imread(str(image))

    rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    face_locations = face_recognition.face_locations(rgb, model='cnn')
    face_encodings = face_recognition.face_encodings(rgb, face_locations)
    for face_encoding in face_encodings:
        known_encodings.append(face_encoding)
        known_names.append(image.parent.name)
    data = {'encodings': known_encodings, 'names': known_names}
return data

```

Η βασική λειτουργία της συνάρτησης είναι η αναγνώριση προσώπων σε μια εικόνα με βάση το σύνολο δεδομένων (dataset). Γίνεται αρχικοποίηση δύο λιστών οι οποίες θα περιέχουν τις κωδικοποιήσεις προσώπου και τα αντίστοιχα ονόματα για το κάθε άτομο του συνόλου δεδομένων και επιστρέφεται μια μη-ταξινομημένη συλλογή τιμών δεδομένων «dict», η οποία θα χρησιμοποιηθεί αργότερα.

Η γραμμή κώδικα παίρνει ως παράμετρο τη διαδρομή προς τον κατάλογο συνόλου δεδομένων.

```
def recognition_faces_image(images_path: Path) -> dict:
```

Δημιουργείται μια λίστα η οποία περιέχει όλες τις εικόνες που θα χρησιμοποιηθούν για την κωδικοποίηση.

```

all_people_images = [image for folder in images_path.iterdir()
                      for image in folder.iterdir()]

```

Απαιτείται η δημιουργία των δύο παρακάτω λιστών.

```
known_encodings = []
```

```
known_names = []
```

Στη συνέχεια, ξεκινάει το loop για τα πρόσωπα των ατόμων.

```
for image in all_people_images:
```

```
    img = cv2.imread(str(image))
```

```
    rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

```
    face_locations = face_recognition.face_locations(rgb, model='cnn')
```

```
    face_encodings = face_recognition.face_encodings(rgb, face_locations)
```

Αυτός ο βρόγχος, για κάθε εικόνα από τις εικόνες προσώπων στο σύνολο δεδομένων, θα φορτώνει την εικόνα «img».

```
img = cv2.imread(str(image))
```

Η OpenCV ορίζει χρωματικά κανάλια στο BGR. Η νέα εικόνα ορίζεται με όνομα μεταβλητής «rgb».

```
rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

Έπειτα, γίνεται εντοπισμός του προσώπου και υπολογίζονται οι κωδικοποιήσεις.

Στις παρακάτω γραμμές κώδικα, εντοπίζονται τα πρόσωπα και παράγεται μια λίστα από boxes προσώπου.

```
face_locations = face_recognition.face_locations(rgb, model='cnn')
```

```
face_encodings = face_recognition.face_encodings(rgb, face_locations)
```

Η μέθοδος «face\_recognition.face\_locations» παίρνει δύο παραμέτρους:

- «rgb»: Η RGB εικόνα.
- «model»: Ορίζεται το μοντέλο εντοπισμού προσώπου που θα χρησιμοποιηθεί. Υπάρχουν δύο επιλογές, οι οποίες είναι: το «HOG» και το «CNN».

Το μοντέλο «HOG» έχει λιγότερη ακρίβεια αλλά είναι πιο γρήγορο με χρήση CPU. Το «CNN» είναι ένα μοντέλο βαθιάς μάθησης με μεγαλύτερη ακρίβεια, το οποίο μπορεί να αποκτήσει μεγαλύτερη ταχύτητα με χρήση GPU/CUDA (στην περίπτωση που υπάρχει η δυνατότητα χρήσης τους).

Η μέθοδος αυτή επιστρέφει έναν πίνακα από οριοθετημένα πλαίσια (boxes) προσώπων τα οποία περιέχονται στην εικόνα.

Η μέθοδος face\_recognition.face\_encodings παίρνει σαν παραμέτρους:

- «rgb»: Η RGB εικόνα.
- «face\_locations»: τα οριοθετημένα πλαίσια κάθε προσώπου, τα οποία προέκυψαν από την προηγούμενη μέθοδο.

Γίνεται μετατροπή των οριοθετημένων κουτιών/πλαισίων του κάθε προσώπου που περιέχεται στην εικόνα, σε μια λίστα 128 αριθμών (διαστάσεων), η διαδικασία αυτή είναι γνωστή και ως κωδικοποίηση του προσώπου σε ένα διάνυσμα.

Από κει και πέρα, πρέπει να προστεθεί η κωδικοποίηση encodings και το όνομα, όπως για παράδειγμα «Will Smith», στην κατάλληλη λίστα (known\_encodings και known\_names). Αυτή η διαδικασία θα εκτελεστεί για όλες τις εικόνες του συνόλου δεδομένων.

Τέλος, θα επιστρέφει μια μη-ταξινομημένη συλλογή τιμών δεδομένων, η οποία περιέχει τα face\_encodings και τα ονόματα ώστε να χρησιμοποιηθεί στο επόμενο σενάριο.

Αυτό γίνεται ως ακολούθως με την εξής συνάρτηση:

```
def recognition_faces_video( video_path: Path, images_path: Path, output_csv_filename: str) ->
None:
    """Recognize faces in a video and output a video.
```

Args:

video\_path (Path): Video Path

images\_path (Path): Images Path

"""

```
people_names = sorted(list(set([image.parent.name for folder in images_path.iterdir() for image
in folder.iterdir()])))
```

```
data = recognition_faces_image(images_path)
```

```
capture_video = cv2.VideoCapture(video_path.name)
```

```
width = int(capture_video.get(cv2.CAP_PROP_FRAME_WIDTH))
```

```
height = int(capture_video.get(cv2.CAP_PROP_FRAME_HEIGHT))
```

```
size = (width, height)
```

```
video_fps = capture_video.get(cv2.CAP_PROP_FPS) # fps of video
```

```
video_codec = cv2.VideoWriter_fourcc(*'MJPG') #fourcc code
```

```
out = cv2.VideoWriter('output_video.avi', video_codec, video_fps, size)
```

```
count_frames = 0
```

```
while True:
```

```
    retval, frame = capture_video.read()
```

```
    if not retval:
```

```
        break
```

```
    rgb_to_bgr = cv2.cvtColor(
```

```
        frame,
```

```
        cv2.COLOR_BGR2RGB
```

```
    )
```

```
    face_locations = face_recognition.face_locations(rgb_to_bgr, model='cnn')
```

```
    face_encodings = face_recognition.face_encodings(rgb_to_bgr, face_locations)
```

```
    names = []
```

```

for face_encoding in face_encodings:
    boolean_matches = face_recognition.compare_faces(data['encodings'],face_encoding,
tolerance=0.6)

    unknown_name = 'unknown'
    if True in boolean_matches:
        matched_idxs = [i for (i, boolean) in enumerate(boolean_matches) if boolean]
        counts = {}
        for i in matched_idxs:
            name = data['names'][i]
            counts[name] = counts.get(name, 0) + 1
            name = max(counts, key=counts.get)
        names.append(name)
    for ((top, right, bottom, left), name) in zip(face_locations, names):

        cv2.rectangle(frame, (left, top),
            (right, bottom), (255, 255, 0), 2)
        y = (top - 15 if top - 15 > 15 else top + 15)
        cv2.putText(frame, name, (left, y),
            cv2.FONT_HERSHEY_SIMPLEX, 0.75, (255, 255, 0), 2)
        append_to_csv(names, count_frames, people_names, output_csv_filename)

    out.write(frame)
    count_frames += 1

capture_video.release()
out.release()
cv2.destroyAllWindows()

```



Στη συνάρτηση `recognition_faces_video` λαμβάνονται οι εξής παράμετροι:

`video_path`: Η διαδρομή του βίντεο που θα υποβληθεί στο σύστημα αναγνώρισης προσώπου.

`images_path`: Η διαδρομή των εικόνων που περιέχει τα πρόσωπα.

`output_csv_file`: Το όνομα του αρχείου csv, όπου θα γραφούν τα αποτελέσματα.

Στις παρακάτω γραμμές κώδικα δημιουργείται μια λίστα, η οποία θα περιέχει το όνομα του κάθε προσώπου από τον κύριο φάκελο. Με την μεταβλητή `data` καλείται η προηγούμενη διαδικασία και επιστρέφει τις προαναφερόμενες τιμές.

```
people_names = sorted(list(set([image.parent.name for folder in images_path.iterdir() for image
in folder.iterdir()])))
data = recognition_faces_image(images_path)
```

Για να εισαχθεί/καταγραφεί ένα βίντεο, πρέπει να δημιουργηθεί ένα αντικείμενο `VideoCapture`. Το όρισμα του οποίου μπορεί να είναι είτε το όνομα ενός αρχείου βίντεο, είτε το ευρετήριο της συσκευής. Αφού ολοκληρωθούν οι διαδικασίες της καταγραφής και της επεξεργασίας του βίντεο `frame-by-frame`, το επόμενο βήμα είναι η αποθήκευση του βίντεο. Για να αποθηκευτεί το επεξεργασμένο βίντεο, πρέπει να δημιουργηθεί ένα αντικείμενο `VideoWriter`.

```
capture_video = cv2.VideoCapture(video_path.name)
width = int(capture_video.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(capture_video.get(cv2.CAP_PROP_FRAME_HEIGHT))
size = (width, height)
video_fps = capture_video.get(cv2.CAP_PROP_FPS) # fps of video
video_codec = cv2.VideoWriter_fourcc(*'MJPG') #fourcc code
out = cv2.VideoWriter('output_video.avi', video_codec, video_fps, size)
```

Παρακάτω περιγράφονται αναλυτικά οι μέθοδοι `VideoCapture()` και `VideoWriter()`:

Η μέθοδος `VideoCapture()` της βιβλιοθήκης `OpenCV` χρησιμοποιείται για να γίνει εισαγωγή/καταγραφή βίντεο με την μορφή αρχείου ή με το άνοιγμα κάμερας για live streaming. Συγκεκριμένα, η μέθοδος παίρνει σαν όρισμα (attribute):

- `video_path`: Τοποθεσία του αρχείου βίντεο στο σύστημα του χρήστη σε μορφή συμβολοσειράς, οι επεκτάσεις του οποίου μπορεί να είναι `.mp4`, `.avi` κ.λ.π. ή
- `device index`: Είναι ένας ακέραιος αριθμός ο οποίος καθορίζει την κάμερα του συστήματος, οι πιθανές τιμές του αριθμού αυτού είναι «0» ή «-1». Σε περίπτωση που είναι συνδεδεμένες στον υπολογιστή περισσότερες από μια κάμερες, τότε μπορεί να γίνει επιλογή της δεύτερης κάμερας εισάγοντας την τιμή «1», αντίστοιχα μπορεί να γίνει επιλογή της τρίτης κάμερας εισάγοντας την τιμή «2» και ούτω καθεξής.

Τέλος επιστρέφει το αντικείμενο καταγραφής βίντεο (frame-by-frame).

```
capture_video = cv2.VideoCapture(video_path.name)
```

Προσοχή, στο τέλος θα πρέπει να γίνει κλείσιμο/απελευθέρωση της καταγραφής! Αυτό επιτυγχάνεται με την εντολή:

```
capture_video.release()
```

Η μέθοδος `VideoWriter()` της βιβλιοθήκης `OpenCV` χρησιμοποιείται για να γίνει αποθήκευση του επεξεργασμένου βίντεο. Θα πρέπει να προσδιοριστεί το όνομα του αρχείου βίντεο το οποίο θα αποθηκευτεί, στην συγκεκριμένη περίπτωση ορίστηκε ως «`output_video.avi`». Έπειτα, πρέπει να προσδιοριστούν ο κώδικας `FourCC`, ο αριθμός των frame ανά δευτερόλεπτο και το μέγεθος των frame.

```
out = cv2.VideoWriter('output_video.avi', video_codec, video_fps, size)
```

Ο `FourCC` είναι ένας κώδικας 4-byte, ο οποίος προσδιορίζει τον κωδικοποιητή του βίντεο. Η λίστα με τους διαθέσιμους κωδικοποιητές μπορεί να βρεθεί διαδικτυακά στον παρακάτω σύνδεσμο:

<https://www.fourcc.org/codecs.php>

Στην συνέχεια ξεκινάει μια διαδικασία επανάληψης, ώστε να βρεθούν και να επεξεργαστούν όλα τα frames του βίντεο. Ο βρόγχος ξεκινάει από την γραμμή κώδικα `while True:` και το πρώτο βήμα είναι να τραβηχτεί ένα frame από την ροή του βίντεο. Στις υπόλοιπες γραμμές του παρακάτω

κώδικα διαβάζεται το frame, επεξεργάζεται και στην συνέχεια εντοπίζονται τα οριοθετημένα πλαίσια προσώπου και υπολογίζονται τα face\_encodings για κάθε οριοθετημένο πλαίσιο.

```
count_frames = 0
```

```
while True:
```

```
    retval, frame = capture_video.read()
```

```
    if not retval:
```

```
        break
```

```
    rgb_to_bgr = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
```

```
    face_locations = face_recognition.face_locations(rgb_to_bgr, model='cnn')
```

```
    face_encodings = face_recognition.face_encodings(rgb_to_bgr, face_locations)
```

Ο παρακάτω κώδικας σχετίζεται με τις κωδικοποιήσεις προσώπου που αφορούν τα πρόσωπα που βρέθηκαν προηγουμένως.

```
names = []
```

```
for face_encoding in face_encodings:
```

```
    boolean_matches = face_recognition.compare_faces(data['encodings'], face_encoding,
tolerance=0.6)
```

```
    unknown_name = 'unknown'
```

```
    if True in boolean_matches:
```

```
        matched_idxs = [i for (i, boolean) in enumerate(boolean_matches) if boolean]
```

```
        counts = { }
```

```
        for i in matched_idxs:
```

```
            name = data['names'][i]
```

```
            counts[name] = counts.get(name, 0) + 1
```

```
        name = max(counts, key=counts.get)
```

```
    names.append(name)
```

Σε αυτό το κομμάτι κώδικα, εισάγεται ένας βρόγχος για κάθε μία από τις κωδικοποιήσεις και επιχειρείται να γίνει σωστή αντιστοιχία των προσώπων.

Η μέθοδος `face_recognition.compare_faces()` της βιβλιοθήκης OpenCV χρησιμοποιείται για να γίνει σύγκριση μιας λίστας κωδικοποιήσεων προσώπου με μια υποψήφια κωδικοποίηση, με σκοπό να βρεθεί αν υπάρχει αντιστοιχία. Οι παράμετροι της μεθόδου είναι οι εξής:

- `known_face_encodings`: Μια λίστα η οποία περιέχει γνωστές κωδικοποιήσεις προσώπου.
- `face_encoding_to_check`: Μια κωδικοποίηση προσώπου η οποία συγκρίνεται με την λίστα.
- `tolerance`: Είναι γνωστή ως «ανοχή» και πρόκειται για την απόσταση μεταξύ των προσώπων έτσι ώστε να θεωρηθεί ότι υπάρχει αντιστοιχία. Όσο πιο μικρή είναι η τιμή της ανοχής τόσο πιο αυστηρή είναι. Η τιμή 0.6 θεωρείται ότι έχει την καλύτερη απόδοση.

Η μέθοδος επιστρέφει μια λίστα από (boolean) τιμές True/False, οι οποίες υποδεικνύουν ποιες από τις γνωστές κωδικοποιήσεις προσώπου έχουν αντιστοιχία με τις προς έλεγχο κωδικοποιήσεις προσώπου.

Στην περίπτωση που βρεθούν αντιστοιχίες, γίνεται καταμέτρηση των πεδίων που επιστρέφουν την τιμή True για κάθε όνομα που βρίσκεται στο σύνολο δεδομένων. Έπειτα, εξάγεται το όνομα το οποίο αντιστοιχεί στο πρόσωπο με τα περισσότερα αληθή (True) αποτελέσματα.

Στο παρακάτω κομμάτι κώδικα, με έναν επαναληπτικό βρόγχο «for» σχεδιάζεται ένα πλαίσιο (box) γύρω από τα πρόσωπα που απεικονίζονται στο βίντεο και εμφανίζεται το αντίστοιχο όνομα πάνω από αυτά.

```
for ((top, right, bottom, left), name) in zip(face_locations, names):
```

```
    cv2.rectangle(frame, (left, top),(right, bottom), (255, 255, 0), 2)
```

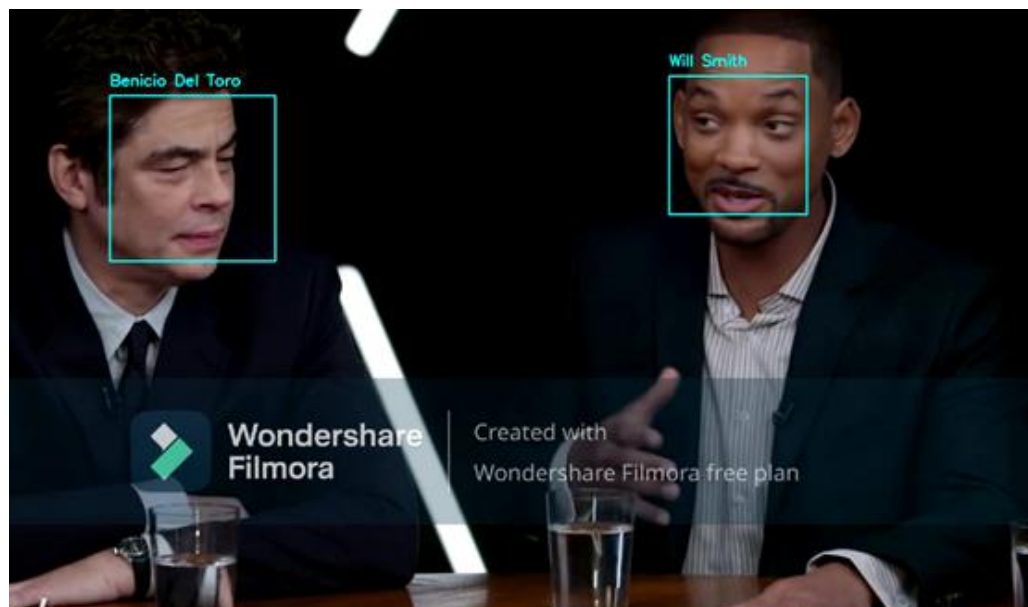
```
    y = (top - 15 if top - 15 > 15 else top + 15)
```

```
    cv2.putText(frame, name, (left, y), cv2.FONT_HERSHEY_SIMPLEX, 0.75, (255, 255, 0), 2)
```

Όταν εκτελεστεί ο κώδικας παράγεται το αποτέλεσμα της Αναγνώρισης προσώπου σε πραγματικό χρόνο.



9. Παράδειγμα 1 - Δημιουργία box γύρω από τα πρόσωπα που απεικονίζονται στο βίντεο και εμφάνιση του αντίστοιχου ονόματος



10. Παράδειγμα 2 - Δημιουργία box γύρω από τα πρόσωπα που απεικονίζονται στο βίντεο και εμφάνιση του αντίστοιχου ονόματος

Για κάθε frame του βίντεο καλείται η εντολή «append\_to\_csv», η οποία εξάγει τα αποτελέσματα σε αρχείο μορφής csv.

```
append_to_csv(names, count_frames, people_names, output_csv_filename)
```

Η συνάρτηση «append\_to\_csv» ορίζεται με τις παρακάτω γραμμές κώδικα.

```
def append_to_csv(people: List[str], frame: int, people_names: List[str], output_csv_filename:
str) -> None:
    seen = [0 for _ in people_names]
    for person in people:
        if person in people_names:
            position_idx = people_names.index(person)
            seen[position_idx] += 1

    results = {
        frame:
            {
                name: see for name, see in zip(people_names, seen)
            }
    }

    file_exists = Path(output_csv_filename).exists()

    with open(output_csv_filename, 'a', newline=") as csv_file:
        fieldnames = ['Frames'] + people_names
        writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
        if not file_exists:
            writer.writeheader()
        for row in results.keys():
            writer.writerow({'Frames': row, **results[row]})
```

Η παραπάνω συνάρτηση δημιουργεί έναν πίνακα με γραμμές και στήλες. Η πρώτη γραμμή αποτελείται από τα Frames και από τα ονόματα των προσώπων που απεικονίζονται στο βίντεο. Για το κάθε πρόσωπο υπάρχει διαφορετική στήλη. Για κάθε frame δημιουργείται μια καινούργια

γραμμή που αποτελείται από τον αριθμό του δεδομένου frame και από τις τιμές «0» ή «1» για το κάθε πρόσωπο, όπου το «0» σημαίνει ότι το πρόσωπο δεν απεικονίζεται στο συγκεκριμένο frame και το «1» σημαίνει ότι το πρόσωπο απεικονίζεται στο συγκεκριμένο frame. Η διαδικασία της επανάληψης ολοκληρώνεται με το τελευταίο frame του βίντεο. Το συγκεκριμένο βίντεο που χρησιμοποιήθηκε ως είσοδος στον κώδικα αποτελείται από 956 frames. Τα αποτελέσματα θα αποθηκευτούν σε ένα αρχείο csv και θα έχουν την παρακάτω μορφή:

Frames	Benicio Del Toro	Joel Edgerton	Mark Ruffalo	Matthew Belloni	Michael Caine	Samuel L. Jackson	Stephen Galloway	Will Smith	unknown	
0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	1	0
5	0	0	0	0	0	0	0	0	1	0
6	0	0	0	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	1	0
10	0	0	0	0	0	0	0	0	1	0
11	0	0	0	0	0	0	0	0	1	0
12	0	0	0	0	0	0	0	0	1	0
13	0	0	0	0	0	0	0	0	1	0
14	0	0	0	0	0	0	0	0	1	0
15	0	0	0	0	0	0	0	0	1	0
16	0	0	0	0	0	0	0	0	1	0
17	0	0	0	0	0	0	0	0	1	0
18	0	0	0	0	0	0	0	0	1	0
19	0	0	0	0	0	0	0	0	1	0
20	0	0	0	0	0	0	0	0	1	0
21	0	0	0	0	0	0	0	0	1	0
22	0	0	0	0	0	0	0	0	1	0
23	0	0	0	0	0	0	0	0	1	0
24	0	0	0	0	0	0	0	0	1	0
25	0	0	0	0	0	0	0	0	1	0
26	0	0	0	0	0	0	0	0	1	0
27	0	0	0	0	0	0	0	0	1	0
28	0	0	0	0	0	0	0	0	1	0
29	0	0	0	0	0	0	0	0	1	0
30	0	0	0	0	0	0	0	0	1	0
31	0	0	0	0	0	0	0	0	1	0
32	0	0	0	0	0	0	0	0	1	0
33	0	0	0	0	0	0	0	0	1	0
34	0	0	0	0	0	0	0	0	1	0
35	0	0	0	0	0	0	0	0	1	0
36	0	0	0	0	0	0	0	0	1	0
37	0	0	0	0	0	0	0	0	1	0
38	0	0	0	0	0	0	0	0	1	0

11. Πίνακας αρχείου CSV που εξάγεται από τον κώδικα

Με την παρακάτω γραμμή κώδικα, μπορεί ο χρήστης να παρακολουθεί την διαδικασία frame by frame έτσι ώστε να γίνει καταγραφή των στοιχείων σύμφωνα με το τι βλέπει πραγματικά ο χρήστης και όχι το τι βλέπει ο κώδικας.

```
cv2_imshow(frame)
```

Αφού έγινε εκτέλεση του κώδικα και εξάχθηκε το τελικό βίντεο, χρησιμοποιήθηκε το πρόγραμμα αναπαραγωγής βίντεο VLC, όπου με κατάλληλες ρυθμίσεις το βίντεο χωρίστηκε σε εικόνες frame by frame. Με τον τρόπο αυτό δημιουργήθηκε χειροκίνητα ένας αντίστοιχος πίνακας με τον προηγούμενο, ο οποίος περιέχει τα αποτελέσματα όπως τα αντιλαμβάνεται ο χρήστης. Ο σκοπός δημιουργίας του πίνακα αυτού είναι για να υπάρχει ένα μέτρο σύγκρισης για τους υπολογισμούς που θα εκτελεστούν παρακάτω. Ακολουθεί παράδειγμα του πίνακα αυτού.

Frames	Benicio Del Toro	Joel Edgerton	Mark Ruffalo	Matthew Belloni	Michael Caine	Samuel L. Jackson	Stephen Galloway	Will Smith	Unknown
0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	1	0
5	0	0	0	0	0	0	0	1	0
6	0	0	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	1	0
10	0	0	0	0	0	0	0	1	0
11	0	0	0	0	0	0	0	1	0
12	0	0	0	0	0	0	0	1	0
13	0	0	0	0	0	0	0	1	0
14	0	0	0	0	0	0	0	1	0
15	0	0	0	0	0	0	0	1	0
16	0	0	0	0	0	0	0	1	0
17	0	0	0	0	0	0	0	1	0
18	0	0	0	0	0	0	0	1	0
19	0	0	0	0	0	0	0	1	0
20	0	0	0	0	0	0	0	1	0
21	0	0	0	0	0	0	0	1	0
22	0	0	0	0	0	0	0	1	0
23	0	0	0	0	0	0	0	1	0
24	0	0	0	0	0	0	0	1	0
25	0	0	0	0	0	0	0	1	0
26	0	0	0	0	0	0	0	1	0
27	0	0	0	0	0	0	0	1	0
28	0	0	0	0	0	0	0	1	0
29	0	0	0	0	0	0	0	1	0
30	0	0	0	0	0	0	0	1	0
31	0	0	0	0	0	0	0	1	0
32	0	0	0	0	0	0	0	1	0
33	0	0	0	0	0	0	0	1	0
34	0	0	0	0	0	0	0	1	0
35	0	0	0	0	0	0	0	1	0
36	0	0	0	0	0	0	0	1	0
37	0	0	0	0	0	0	0	1	0
38	0	0	0	0	0	0	0	1	0
39	0	0	0	0	0	0	0	1	0
40	0	0	0	0	0	0	0	1	0

12. Πίνακας "Actual Truth"



Ο κώδικας εκτελέστηκε συνολικά 15 φορές με διαφορετική τιμή «ανοχής»(tolerance) και με διαφορετικό αριθμό φωτογραφιών των προσώπων των ατόμων που αποτελούν το dataset.

Περιγράφεται αναλυτικά η διαδικασία που ακολουθήθηκε:

- Πείραμα 1: Εκτελέστηκε ο κώδικας με tolerance=0.4 και dataset το οποίο αποτελείται από 10 φωτογραφίες για το κάθε άτομο.
- Πείραμα 2: Εκτελέστηκε ο κώδικας με tolerance=0.5 και dataset το οποίο αποτελείται από 10 φωτογραφίες για το κάθε άτομο.
- Πείραμα 3: Εκτελέστηκε ο κώδικας με tolerance=0.6 και dataset το οποίο αποτελείται από 10 φωτογραφίες για το κάθε άτομο.
- Πείραμα 4: Εκτελέστηκε ο κώδικας με tolerance=0.7 και dataset το οποίο αποτελείται από 10 φωτογραφίες για το κάθε άτομο.
- Πείραμα 5: Εκτελέστηκε ο κώδικας με tolerance=0.8 και dataset το οποίο αποτελείται από 10 φωτογραφίες για το κάθε άτομο.
- Πείραμα 6: Εκτελέστηκε ο κώδικας με tolerance=0.4 και dataset το οποίο αποτελείται από 20 φωτογραφίες για το κάθε άτομο.
- Πείραμα 7: Εκτελέστηκε ο κώδικας με tolerance=0.5 και dataset το οποίο αποτελείται από 20 φωτογραφίες για το κάθε άτομο.
- Πείραμα 8: Εκτελέστηκε ο κώδικας με tolerance=0.6 και dataset το οποίο αποτελείται από 20 φωτογραφίες για το κάθε άτομο.
- Πείραμα 9: Εκτελέστηκε ο κώδικας με tolerance=0.7 και dataset το οποίο αποτελείται από 20 φωτογραφίες για το κάθε άτομο.
- Πείραμα 10: Εκτελέστηκε ο κώδικας με tolerance=0.8 και dataset το οποίο αποτελείται από 20 φωτογραφίες για το κάθε άτομο.
- Πείραμα 11: Εκτελέστηκε ο κώδικας με tolerance=0.4 και dataset το οποίο αποτελείται από 30 φωτογραφίες για το κάθε άτομο.
- Πείραμα 12: Εκτελέστηκε ο κώδικας με tolerance=0.5 και dataset το οποίο αποτελείται από 30 φωτογραφίες για το κάθε άτομο.
- Πείραμα 13: Εκτελέστηκε ο κώδικας με tolerance=0.6 και dataset το οποίο αποτελείται από 30 φωτογραφίες για το κάθε άτομο.
- Πείραμα 14: Εκτελέστηκε ο κώδικας με tolerance=0.7 και dataset το οποίο αποτελείται από 30 φωτογραφίες για το κάθε άτομο.
- Πείραμα 15: Εκτελέστηκε ο κώδικας με tolerance=0.8 και dataset το οποίο αποτελείται από 30 φωτογραφίες για το κάθε άτομο.

Για το κάθε πείραμα γίνεται υπολογισμός των ακολούθων μοντέλων ταξινόμησης:

- True Positive: Είναι ένα αποτέλεσμα, όπου το μοντέλο προβλέπει σωστά την θετική κλάση (True ή 1).
- True Negative: Είναι ένα αποτέλεσμα, όπου το μοντέλο προβλέπει σωστά την αρνητική κλάση (False ή 0).
- False Positive: Είναι ένα αποτέλεσμα, όπου το μοντέλο προβλέπει εσφαλμένα την θετική κλάση (True ή 1).
- False Negative: Είναι ένα αποτέλεσμα, όπου το μοντέλο προβλέπει εσφαλμένα την αρνητική κλάση (False ή 0).

Η δομή του αρχείου excel που περιέχονται όλα τα παραπάνω πειράματα είναι η εξής:

1. Actual Truth (ο πίνακας που δημιουργήθηκε χειροκίνητα)
2. Πείραμα 1
3. Πείραμα 2
4. Πείραμα 3
5. Πείραμα 4
6. Πείραμα 5
7. Πείραμα 6
8. Πείραμα 7
9. Πείραμα 8
10. Πείραμα 9
11. Πείραμα 10
12. Πείραμα 11
13. Πείραμα 12
14. Πείραμα 13
15. Πείραμα 14
16. Πείραμα 15
17. Αποτελέσματα

Το πρώτο λογιστικό φύλλο με ονομασία «Actual Truth», αποτελείται από τα στοιχεία που έχουν καταγραφεί χειροκίνητα. Στο δεύτερο λογιστικό φύλλο εισάγονται τα αποτελέσματα, τα οποία εξάχθηκαν από τον κώδικα για το «Πείραμα 1». Στη συνέχεια δημιουργείται για το κάθε πιθανό αποτέλεσμα ένας πίνακας και ξεκινούν οι υπολογισμοί με την βοήθεια συναρτήσεων που μας παρέχονται από το Excel.

Για τον υπολογισμό των True Positive, ακολουθείται η λογική ότι εάν το αποτέλεσμα που βλέπει ο χρήστης είναι True (1) και το αποτέλεσμα που εξάγει ο κώδικας είναι True (1), τότε είναι True

Positive και η τιμή που παίρνει είναι «1», αλλιώς «0». Αυτό εκτελείται για ολόκληρη την στήλη του κάθε ονόματος.

Εισάγονται οι εξής συναρτήσεις για τη δημιουργία των στηλών των ατόμων του Πίνακα των True Positives:

```
=IF((AND(Actual_Truth!B2 = Peirama1!B2,Actual_Truth!B2 =1)), 1, 0)  
=IF((AND(Actual_Truth!C2 = Peirama1!C2,Actual_Truth!C2 =1)), 1, 0)  
=IF((AND(Actual_Truth!D2 = Peirama1!D2,Actual_Truth!D2 =1)), 1, 0)  
=IF((AND(Actual_Truth!E2 = Peirama1!E2,Actual_Truth!E2 =1)), 1, 0)  
=IF((AND(Actual_Truth!F2 = Peirama1!F2,Actual_Truth!F2 =1)), 1, 0)  
=IF((AND(Actual_Truth!G2 = Peirama1!G2,Actual_Truth!G2 =1)), 1, 0)  
=IF((AND(Actual_Truth!H2 = Peirama1!H2,Actual_Truth!H2 =1)), 1, 0)  
=IF((AND(Actual_Truth!I2 = Peirama1!I2,Actual_Truth!I2 =1)), 1, 0)
```

Στο τέλος του πίνακα υπολογίζεται το σύνολο των True Positives με σκοπό να χρησιμοποιηθεί αργότερα στον υπολογισμό των ποσοστών accuracy, precision, recall και F1 score.

Για την εύρεση των True (1) της κάθε στήλης χρησιμοποιούνται οι εξής συναρτήσεις:

```
=COUNTIF(N2:N957,1)  
=COUNTIF(O2:O957,1)  
=COUNTIF(P2:P957,1)  
=COUNTIF(Q2:P957,1)  
=COUNTIF(R2:P957,1)  
=COUNTIF(S2:P957,1)  
=COUNTIF(T2:P957,1)  
=COUNTIF(U2:P957,1)
```

Για την εύρεση του συνολικού αθροίσματος του Πίνακα των True Positives χρησιμοποιείται η εξής συνάρτηση:

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΣΩΠΩΝ ΣΕ ΒΙΝΤΕΟ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ

=SUM(N958:U958)

Frame	Ben	Joe	Ma	Ma	Mid	San	Ste	Wi	unt	Frames	Benicio D	Joel Edge	Mark Ruff	Matthew	Michael C	Samuel L	Stephen C	Will Smith	Unknown
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	1
3	0	0	0	0	0	0	0	0	1	0	3	0	0	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0	1	0	4	0	0	0	0	0	0	0	1
5	0	0	0	0	0	0	0	0	1	0	5	0	0	0	0	0	0	0	1
6	0	0	0	0	0	0	0	0	1	0	6	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	0	1	0	7	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	1	0	8	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	1	0	9	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	1	0	10	0	0	0	0	0	0	0	1
11	0	0	0	0	0	0	0	0	1	0	11	0	0	0	0	0	0	0	1
12	0	0	0	0	0	0	0	0	1	0	12	0	0	0	0	0	0	0	1
13	0	0	0	0	0	0	0	0	1	0	13	0	0	0	0	0	0	0	1
14	0	0	0	0	0	0	0	0	1	0	14	0	0	0	0	0	0	0	1
15	0	0	0	0	0	0	0	0	1	0	15	0	0	0	0	0	0	0	1
16	0	0	0	0	0	0	0	0	1	0	16	0	0	0	0	0	0	0	1
17	0	0	0	0	0	0	0	0	1	0	17	0	0	0	0	0	0	0	1
18	0	0	0	0	0	0	0	0	1	0	18	0	0	0	0	0	0	0	1
19	0	0	0	0	0	0	0	0	1	0	19	0	0	0	0	0	0	0	1
20	0	0	0	0	0	0	0	0	1	0	20	0	0	0	0	0	0	0	1
21	0	0	0	0	0	0	0	0	1	0	21	0	0	0	0	0	0	0	1
22	0	0	0	0	0	0	0	0	1	0	22	0	0	0	0	0	0	0	1
23	0	0	0	0	0	0	0	0	1	0	23	0	0	0	0	0	0	0	1
24	0	0	0	0	0	0	0	0	1	0	24	0	0	0	0	0	0	0	1
25	0	0	0	0	0	0	0	0	1	0	25	0	0	0	0	0	0	0	1
26	0	0	0	0	0	0	0	0	1	0	26	0	0	0	0	0	0	0	1
27	0	0	0	0	0	0	0	0	1	0	27	0	0	0	0	0	0	0	1
28	0	0	0	0	0	0	0	0	1	0	28	0	0	0	0	0	0	0	1
29	0	0	0	0	0	0	0	0	1	0	29	0	0	0	0	0	0	0	1
30	0	0	0	0	0	0	0	0	1	0	30	0	0	0	0	0	0	0	1
...											...								
940	0	0	0	0	0	0	0	0	0	0	940	0	0	0	0	0	0	0	0
941	0	0	0	0	0	0	0	0	0	0	941	0	0	0	0	0	0	0	0
942	0	0	0	0	0	0	0	0	0	0	942	0	0	0	0	0	0	0	0
943	0	0	0	0	0	0	0	0	0	0	943	0	0	0	0	0	0	0	0
944	0	0	0	0	0	0	0	0	0	0	944	0	0	0	0	0	0	0	0
945	0	0	0	0	0	0	0	0	0	0	945	0	0	0	0	0	0	0	0
946	0	0	0	0	0	0	0	0	0	0	946	0	0	0	0	0	0	0	0
947	0	0	0	0	0	0	0	0	0	0	947	0	0	0	0	0	0	0	0
948	0	0	0	0	0	0	0	0	0	0	948	0	0	0	0	0	0	0	0
949	0	0	0	0	0	0	0	0	0	0	949	0	0	0	0	0	0	0	0
950	0	0	0	0	0	0	0	0	0	0	950	0	0	0	0	0	0	0	0
951	0	0	0	0	0	0	0	0	0	0	951	0	0	0	0	0	0	0	0
952	0	0	0	0	0	0	0	0	0	0	952	0	0	0	0	0	0	0	0
953	0	0	0	0	0	0	0	0	0	0	953	0	0	0	0	0	0	0	0
954	0	0	0	0	0	0	0	0	0	0	954	0	0	0	0	0	0	0	0
955	0	0	0	0	0	0	0	0	0	0	955	0	0	0	0	0	0	0	0
											TP	336	26	75	69	49	46	90	513
											SUM								
											1204								

13. Παράδειγμα Πίνακα "TRUE POSITIVES"

Εκτελείται η ίδια διαδικασία για τον υπολογισμό των True Negative, ακολουθείται η λογική ότι εάν το αποτέλεσμα που βλέπει ο χρήστης είναι False (0) και το αποτέλεσμα που εξάγει ο κώδικας είναι False (0), τότε είναι True Negative και η τιμή που παίρνει είναι «1», αλλιώς «0». Αυτό εκτελείται για ολόκληρη την στήλη του κάθε ονόματος.

Εισάγονται οι εξής συναρτήσεις για την δημιουργία των στηλών των ατόμων του Πίνακα των True Negatives:

=IF((AND(Actual\_Truth!B2 = Peirama1!B2,Actual\_Truth!B2 =0)), 1, 0)

=IF((AND(Actual\_Truth!C2 = Peirama1!C2,Actual\_Truth!C2 =0)), 1, 0)

=IF((AND(Actual\_Truth!D2 = Peirama1!D2,Actual\_Truth!D2 =0)), 1, 0)

=IF((AND(Actual\_Truth!E2 = Peirama1!E2,Actual\_Truth!E2 =0)), 1, 0)

=IF((AND(Actual\_Truth!F2 = Peirama1!F2,Actual\_Truth!F2 =0)), 1, 0)

=IF((AND(Actual\_Truth!G2 = Peirama1!G2,Actual\_Truth!G2 =0)), 1, 0)

=IF((AND(Actual\_Truth!H2 = Peirama1!H2,Actual\_Truth!H2 =0)), 1, 0)

=IF((AND(Actual\_Truth!I2 = Peirama1!I2,Actual\_Truth!I2 =0)), 1, 0)

Στο τέλος του πίνακα υπολογίζεται το σύνολο των True Negatives με σκοπό να χρησιμοποιηθεί αργότερα στον υπολογισμό των ποσοστών accuracy, precision, recall και F1 score.

Για την εύρεση των True (1) της κάθε στήλης χρησιμοποιούνται οι εξής συναρτήσεις:

=COUNTIF(Z2:Z957,1)

=COUNTIF(AA2:AA957,1)

=COUNTIF(AB2:AB957,1)

=COUNTIF(AC2:AC957,1)

=COUNTIF(AD2:AD957,1)

=COUNTIF(AE2:AE957,1)

=COUNTIF(AF2:AF957,1)

=COUNTIF(AG2:AG957,1)

Για την εύρεση του συνολικού αθροίσματος του Πίνακα των True Negatives χρησιμοποιείται η εξής συνάρτηση:

=SUM(Z958:AG958)

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΣΩΠΩΝ ΣΕ ΒΙΝΤΕΟ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ

940	1	1	1	1	1	1	1	1
941	1	1	1	1	1	1	1	1
942	1	1	1	1	1	1	1	1
943	1	1	1	1	1	1	1	1
944	1	1	1	1	1	1	1	1
945	1	1	1	1	1	1	1	1
946	1	1	1	1	1	1	1	1
947	1	1	1	1	1	1	1	1
948	1	1	1	1	1	1	1	1
949	1	1	1	1	1	1	1	1
950	1	1	1	1	1	1	1	1
951	1	1	1	1	1	1	1	1
952	1	1	1	1	1	1	1	1
953	1	1	1	1	1	1	1	1
954	1	1	1	1	1	1	1	1
955	1	1	1	1	1	1	1	1
TN	575	907	881	769	907	910	769	410
SUM								
6128								

14. Παράδειγμα Πίνακα "TRUE NEGATIVES"

Εκτελείται η ίδια διαδικασία για τον υπολογισμό των False Positive, ακολουθείται η λογική ότι εάν το αποτέλεσμα που βλέπει ο χρήστης είναι False (0) και το αποτέλεσμα που εξάγει ο κώδικας είναι True (1), τότε είναι False Positive και η τιμή που παίρνει είναι «1», αλλιώς «0». Αυτό εκτελείται για ολόκληρη την στήλη του κάθε ονόματος.

Εισάγονται οι εξής συναρτήσεις για την δημιουργία των στηλών των ατόμων του Πίνακα των False Positives:

=IF((AND(Actual\_Truth!B2=0,Peirama1!B2=1)), 1, 0)

=IF((AND(Actual\_Truth!C2=0,Peirama1!C2=1)), 1, 0)

=IF((AND(Actual\_Truth!D2=0,Peirama1!D2=1)), 1, 0)

=IF((AND(Actual\_Truth!E2=0,Peirama1!E2=1)), 1, 0)

=IF((AND(Actual\_Truth!F2=0,Peirama1!F2=1)), 1, 0)

=IF((AND(Actual\_Truth!G2=0,Peirama1!G2=1)), 1, 0)

=IF((AND(Actual\_Truth!H2=0,Peirama1!H2=1)), 1, 0)

=IF((AND(Actual\_Truth!I2=0,Peirama1!I2=1)), 1, 0)

Στο τέλος του πίνακα υπολογίζεται το σύνολο των False Positives με σκοπό να χρησιμοποιηθεί αργότερα στον υπολογισμό των ποσοστών accuracy, precision, recall και F1 score.

Για την εύρεση των True (1) της κάθε στήλης χρησιμοποιούνται οι εξής συναρτήσεις:

- =COUNTIF(AL2:AL957,1)
- =COUNTIF(AM2:AM957,1)
- =COUNTIF(AN2:AN957,1)
- =COUNTIF(AO2:AO957,1)
- =COUNTIF(AP2:AP957,1)
- =COUNTIF(AQ2:AQ957,1)
- =COUNTIF(AR2:AR957,1)
- =COUNTIF(AS2:AS957,1)

Για την εύρεση του συνολικού αθροίσματος του Πίνακα των False Positives χρησιμοποιείται η εξής συνάρτηση:

- =SUM(AL958:AS958)

940	0	0	0	0	0	0	0	0
941	0	0	0	0	0	0	0	0
942	0	0	0	0	0	0	0	0
943	0	0	0	0	0	0	0	0
944	0	0	0	0	0	0	0	0
945	0	0	0	0	0	0	0	0
946	0	0	0	0	0	0	0	0
947	0	0	0	0	0	0	0	0
948	0	0	0	0	0	0	0	0
949	0	0	0	0	0	0	0	0
950	0	0	0	0	0	0	0	0
951	0	0	0	0	0	0	0	0
952	0	0	0	0	0	0	0	0
953	0	0	0	0	0	0	0	0
954	0	0	0	0	0	0	0	0
955	0	0	0	0	0	0	0	0
FP	0	0	0	0	0	0	0	0
SUM								
0								

15. Παράδειγμα Πίνακα "FALSE POSITIVES"

Εκτελείται η ίδια διαδικασία για τον υπολογισμό των False Negative, ακολουθείται η λογική ότι εάν το αποτέλεσμα που βλέπει ο χρήστης είναι True (1) και το αποτέλεσμα που εξάγει ο κώδικας

είναι False (0), τότε είναι False Negative και η τιμή που παίρνει είναι «1», αλλιώς «0». Αυτό εκτελείται για ολόκληρη την στήλη του κάθε ονόματος.

Εισάγονται οι εξής συναρτήσεις για την δημιουργία των στηλών των ατόμων του Πίνακα των False Negatives:

=IF((AND(Actual\_Truth!B2=1,Peirama1!B2=0)), 1, 0)

=IF((AND(Actual\_Truth!C2=1,Peirama1!C2=0)), 1, 0)

=IF((AND(Actual\_Truth!D2=1,Peirama1!D2=0)), 1, 0)

=IF((AND(Actual\_Truth!E2=1,Peirama1!E2=0)), 1, 0)

=IF((AND(Actual\_Truth!F2=1,Peirama1!F2=0)), 1, 0)

=IF((AND(Actual\_Truth!G2=1,Peirama1!G2=0)), 1, 0)

=IF((AND(Actual\_Truth!H2=1,Peirama1!H2=0)), 1, 0)

=IF((AND(Actual\_Truth!I2=1,Peirama1!I2=0)), 1, 0)

Στο τέλος του πίνακα υπολογίζεται το σύνολο των False Negatives με σκοπό να χρησιμοποιηθεί αργότερα στον υπολογισμό των ποσοστών accuracy, precision, recall και F1 score.

Για την εύρεση των True (1) της κάθε στήλης χρησιμοποιούνται οι εξής συναρτήσεις:

=COUNTIF(AX2:AX957,1)

=COUNTIF(AY2:AY957,1)

=COUNTIF(AZ2:AZ957,1)

=COUNTIF(BA2:BA957,1)

=COUNTIF(BB2:BB957,1)

=COUNTIF(BC2:BC957,1)

=COUNTIF(BD2:BD957,1)

=COUNTIF(BE2:BE957,1)

Για την εύρεση του συνολικού αθροίσματος του Πίνακα των False Negatives χρησιμοποιείται η εξής συνάρτηση:



ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΣΩΠΩΝ ΣΕ ΒΙΝΤΕΟ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ

=SUM(AX958:BE958)

940	0	0	0	0	0	0	0	0
941	0	0	0	0	0	0	0	0
942	0	0	0	0	0	0	0	0
943	0	0	0	0	0	0	0	0
944	0	0	0	0	0	0	0	0
945	0	0	0	0	0	0	0	0
946	0	0	0	0	0	0	0	0
947	0	0	0	0	0	0	0	0
948	0	0	0	0	0	0	0	0
949	0	0	0	0	0	0	0	0
950	0	0	0	0	0	0	0	0
951	0	0	0	0	0	0	0	0
952	0	0	0	0	0	0	0	0
953	0	0	0	0	0	0	0	0
954	0	0	0	0	0	0	0	0
955	0	0	0	0	0	0	0	0
FN	45	23	0	118	0	0	97	33
SUM								
316								

16. Παράδειγμα Πίνακα "FALSE NEGATIVES"

Αφού ολοκληρωθεί η δημιουργία των παραπάνω πινάκων μπορούν να υπολογιστούν τα ποσοστά για την αξιολόγηση της απόδοσης των μοντέλων ταξινόμησης. Συγκεκριμένα τα ποσοστά που θα υπολογιστούν για το κάθε πείραμα είναι τα εξής:

- Accuracy: Η «ακρίβεια» ορίζεται ως ο λόγος των σωστά ταξινομημένων περιπτώσεων δεδομένων προς τον συνολικό αριθμό περιπτώσεων δεδομένων.

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

- Precision: Η «ακρίβεια» (θετική προγνωστική αξία) ορίζεται ως ο λόγος των πραγματικών θετικών προς το άθροισμα των πραγματικών θετικών και των εσφαλμένων θετικών. Η ιδανική περίπτωση είναι να είναι ίση με την τιμή «1». Αυτό συμβαίνει μόνο όταν ο αριθμητής και ο παρονομαστής είναι ίσοι, δηλαδή  $TP = TP + FP$ , πράγμα το οποίο σημαίνει ότι το  $FP$  είναι μηδέν. Στην περίπτωση που το  $FP$  αυξάνεται, ο παρονομαστής γίνεται μεγαλύτερος από τον αριθμητή, με αποτέλεσμα η ακρίβεια να μειώνεται.

$$Precision = \frac{TP}{TP + FP}$$

- Recall: Ορίζεται ως ο λόγος των πραγματικών θετικών προς το άθροισμα των πραγματικών θετικών και των εσφαλμένων αρνητικών. Είναι γνωστό και ως «ευαισθησία» ή «πραγματικός θετικός ρυθμός». Η ιδανική περίπτωση είναι να είναι ίση με την τιμή «1». Αυτό συμβαίνει μόνο όταν ο αριθμητής και ο παρονομαστής είναι ίσοι, δηλαδή  $TP = TP + FN$ , πράγμα το οποίο σημαίνει ότι το  $FN$  είναι μηδέν. Στην περίπτωση που το  $FN$  αυξάνεται, ο παρονομαστής γίνεται μεγαλύτερος από τον αριθμητή, με αποτέλεσμα η ακρίβεια να μειώνεται.

$$Recall = \frac{TP}{TP + FN}$$

- F1 Score: Είναι ο μέσος όρος της «ακρίβειας» (precision) και του «πραγματικού θετικού ρυθμού» (recall). Ισούται με την τιμή «1» μόνο όταν και το precision και το recall είναι «1». Το F1 Score αυξάνεται μόνο όταν το precision και το recall αυξάνονται. Ορίζεται ως αρμονικός μέσος όρος του precision και του recall και είναι καλύτερο μετρικό σύστημα από την ακρίβεια (accuracy).

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Τα αποτελέσματα για το κάθε πείραμα παρουσιάζονται παρακάτω:

Πείραμα 1 - Πείραμα 5, περιέχουν 10 φωτογραφίες για το κάθε άτομο					
Tolerance	0.4	0.5	0.6	0.7	0.8
	Πείραμα 1	Πείραμα 2	Πείραμα 3	Πείραμα 4	Πείραμα 5
Accuracy	95.87%	99.25%	100%	99.96%	92.01%
Precision	100.00%	100%	100%	100%	80.14%
Recall	79.21%	96.25%	100%	99.80%	78.80%
F1 Score	88.40%	98.09%	100%	99.90%	79.46%

Πείραμα 6 - Πείραμα 10, περιέχουν 20 φωτογραφίες για το κάθε άτομο					
Tolerance	0.4	0.5	0.6	0.7	0.8
	Πείραμα 6	Πείραμα 7	Πείραμα 8	Πείραμα 9	Πείραμα 10
Accuracy	96.22%	99.27%	100%	99.99%	94.58%
Precision	100%	100%	100%	100%	87.22%
Recall	80.99%	96.32%	100%	99.93%	84.38%
F1 Score	89.49%	98.12%	100%	99.97%	85.77%

Πείραμα 11 - Πείραμα 15, περιέχουν 30 φωτογραφίες για το κάθε άτομο					
Tolerance	0.4	0.5	0.6	0.7	0.8
	Πείραμα 11	Πείραμα 12	Πείραμα 13	Πείραμα 14	Πείραμα 15
Accuracy	96.59%	99.23%	100%	99.99%	94.87%
Precision	100%	99.80%	100%	100%	88.06%
Recall	82.83%	96.32%	100%	99.93%	84.95%
F1 Score	90.61%	98.02%	100%	99.97%	86.48%

17. Αποτελέσματα Πειραμάτων

Παρατηρείται ότι τα πειράματα 3, 8 και 13 με tolerance=0.6 έχουν ποσοστά accuracy, precision, recall και F1 score ίσα με 100%, ανεξάρτητα από το πλήθος φωτογραφιών που εισάχθηκαν στο dataset. Αυτό σημαίνει ότι το 0.6 είναι το ιδανικότερο.

Στα πειράματα 1, 6 και 11 με tolerance=0.4 παρατηρείται αύξηση των ποσοστών καθώς εισάγονται περισσότερες φωτογραφίες στο dataset. Ενώ, στα πειράματα 5, 10 και 15 παρατηρείται μείωση των ποσοστών καθώς εισάγονται περισσότερες φωτογραφίες στο dataset

Συμπερασματικά, οι ακραίες τιμές tolerance=0.4 και αντίστοιχα tolerance=0.8 παρουσιάζουν μικρότερα ποσοστά accuracy, precision, recall και F1 score σε σύγκριση με την τιμή tolerance=0.6 που αποδείχθηκε ως η πιο αποτελεσματική.

## Συμπεράσματα

Η ανίχνευση προσώπου είναι ένα πρόβλημα όρασης στον υπολογιστή που περιλαμβάνει την εύρεση προσώπων σε φωτογραφίες.

Ο εντοπισμός ενός προσώπου σε μια φωτογραφία αναφέρεται στην εύρεση της συντεταγμένης του προσώπου στην εικόνα, ενώ ο εντοπισμός αναφέρεται στην οριοθέτηση της έκτασης του προσώπου, συχνά μέσω ενός πλαισίου οριοθέτησης γύρω από το πρόσωπο.

Η ανίχνευση προσώπων σε μια φωτογραφία λύνεται εύκολα από τους ανθρώπους, αν και ιστορικά υπήρξε πρόκληση για τους υπολογιστές, δεδομένης της δυναμικής φύσης των προσώπων. Για παράδειγμα, τα πρόσωπα πρέπει να ανιχνεύονται ανεξάρτητα από τον προσανατολισμό ή τη γωνία που αντιμετωπίζουν, τα επίπεδα φωτός, τα ρούχα, τα αξεσουάρ, το χρώμα των μαλλιών, τα μαλλιά του προσώπου, το μακιγιάζ, την ηλικία και ούτω καθεξής.

Με μια φωτογραφία, ένα σύστημα ανίχνευσης προσώπου θα εξάγει μηδέν ή περισσότερα κουτιά οριοθέτησης που περιέχουν πρόσωπα. Τα ανιχνευμένα πρόσωπα μπορούν στη συνέχεια να παρέχονται ως είσοδος σε ένα επόμενο σύστημα, όπως ένα σύστημα αναγνώρισης προσώπου.

Υπάρχουν ίσως δύο βασικές προσεγγίσεις για την αναγνώριση προσώπων: μέθοδοι βασισμένες σε χαρακτηριστικά που χρησιμοποιούν χειροποίητα φίλτρα για αναζήτηση και ανίχνευση προσώπων και μεθόδους βασισμένες σε εικόνα που μαθαίνουν ολιστικά πώς να εξαγάγουμε πρόσωπα από ολόκληρη την εικόνα.

Είναι ένα ασήμαντο πρόβλημα για τον άνθρωπο να λυθεί και έχει λυθεί αρκετά καλά με κλασικές τεχνικές βασισμένες σε χαρακτηριστικά, όπως ο ταξινομητής καταρράκτη. Πιο πρόσφατα, οι μέθοδοι βαθιάς μάθησης έχουν επιτύχει προηγμένα αποτελέσματα σε τυπικά σύνολα δεδομένων ανίχνευσης προσώπου. Ένα παράδειγμα είναι το Multi-task Cascade Convolutional Neural Network ή το MTCNN για συντομία.

## Βιβλιογραφία

- [1] A. Turing, *Intelligent machinery (1948)*. 2004.
- [2] A. M. TURING, “I.—COMPUTING MACHINERY AND INTELLIGENCE,” *Mind*, vol. LIX, no. 236, pp. 433–460, Oct. 1950, doi: 10.1093/mind/LIX.236.433.
- [3] A. M. Turing, “1. the imitation game,” in *Theories of Mind: An Introductory Reader*, M. Eckert, Ed. Rowman & Littlefield, 2006.
- [4] M. Neelam, *Neelam MahaLakshmi (2021) Aspects of Artificial Intelligence In Karthikeyan.J, Su-Hie Ting and Yu-Jin Ng (eds), “Learning Outcomes of Classroom Research” p:250-256, L’ Ordine Nuovo Publication, India. 978-93-92995-15-6*. 2022.
- [5] J. R. Anderson, *Machine learning: An artificial intelligence approach*, vol. 3. Morgan Kaufmann, 1990.
- [6] P. Ongsulee, “Artificial intelligence, machine learning and deep learning,” in *2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE)*, 2017, pp. 1–6. doi: 10.1109/ICTKE.2017.8259629.
- [7] O. Campesato, *Artificial intelligence, machine learning, and deep learning*. Dulles, Virginia ; Mercury Learning and Information, 2020.
- [8] A. Abraham, “Artificial Neural Networks,” in *Handbook of Measuring System Design*, Chichester, UK: John Wiley & Sons, Ltd, 2005. doi: 10.1002/0471497398.mm421.
- [9] L. Deng, “Deep Learning: Methods and Applications,” *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014, doi: 10.1561/20000000039.
- [10] J. Ahmad, H. Farman, and Z. Jan, “Deep Learning Methods and Applications,” 2019, pp. 31–42. doi: 10.1007/978-981-13-3459-7\_3.
- [11] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [12] H. Liu and B. Lang, “Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey,” *Applied Sciences*, vol. 9, no. 20, p. 4396, Oct. 2019, doi: 10.3390/app9204396.
- [13] Y. Xin *et al.*, “Machine Learning and Deep Learning Methods for Cybersecurity,” *IEEE Access*, vol. 6, pp. 35365–35381, 2018, doi: 10.1109/ACCESS.2018.2836950.

- [14] W. Wang and K. Siau, “Artificial Intelligence, Machine Learning, Automation, Robotics, Future of Work and Future of Humanity: A Review and Research Agenda,” *Journal of Database Management*, vol. 30, pp. 61–79, Jan. 2019, doi: 10.4018/JDM.2019010104.
- [15] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, Aug. 2017, pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.
- [16] M. Coşkun, A. Uçar, Ö. Yildirim, and Y. Demir, “Face recognition based on convolutional neural network,” in *2017 International Conference on Modern Electrical and Energy Systems (MEES)*, 2017, pp. 376–379. doi: 10.1109/MEES.2017.8248937.
- [17] H. Singh, *Practical Machine Learning and Image Processing: For Facial Recognition, Object Detection, and Pattern Recognition Using Python*, 1st ed. APress, 2019.
- [18] K. Aljafaar, “Face recognition using Viola-Jones depending on Python,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, pp. 1513–1521, Dec. 2020, doi: 10.11591/ijeecs.v20.i3.pp1513-1521.
- [19] M. Da’san, A. Alqudah, and O. Debeir, “Face detection using Viola and Jones method and neural networks,” in *2015 International Conference on Information and Communication Technology Research (ICTRC)*, May 2015, pp. 40–43. doi: 10.1109/ICTRC.2015.7156416.
- [20] R. R. Damanik, D. Sitanggang, H. Pasaribu, H. Siagian, and F. Gulo, “An application of viola jones method for face recognition for absence process efficiency,” *Journal of Physics: Conference Series*, vol. 1007, p. 012013, Apr. 2018, doi: 10.1088/1742-6596/1007/1/012013.
- [21] K. PNithish Sriman, P. Raj Kumar, A. Naveen, and R. Saravana Kumar, “Comparison of Paul Viola – Michael Jones algorithm and HOG algorithm for Face Detection,” *IOP Conference Series: Materials Science and Engineering*, vol. 1084, no. 1, p. 012014, 2021, doi: 10.1088/1757-899x/1084/1/012014.
- [22] K. Vikram and S. Padmavathi, “Facial parts detection using Viola Jones algorithm,” in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Jan. 2017, pp. 1–4. doi: 10.1109/ICACCS.2017.8014636.
- [23] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001, vol. 1, pp. I–I. doi: 10.1109/CVPR.2001.990517.

- [24] P. Viola and M. Jones, “Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade,” in *Advances in Neural Information Processing Systems*, 2001, vol. 14. [Online]. Available: <https://proceedings.neurips.cc/paper/2001/file/0b1ec366924b26fc98fa7b71a9c249cf-Paper.pdf>
- [25] P. I. Wilson and J. Fernandez, “Facial Feature Detection Using Haar Classifiers,” *J. Comput. Sci. Coll.*, vol. 21, no. 4, pp. 127–133, Apr. 2006.
- [26] Hoang Minh Phuong, le Dung, T. de Souza-Daw, Nguyen Tien Dzung, and Thang Manh Hoang, “Extraction of human facial features based on Haar feature with Adaboost and image recognition techniques,” in *2012 Fourth International Conference on Communications and Electronics (ICCE)*, Aug. 2012, pp. 302–305. doi: 10.1109/CCE.2012.6315916.
- [27] D. Peleshko and K. Soroka, “Research of usage of Haar-like features and AdaBoost algorithm in Viola-Jones method of object detection,” in *2013 12th International Conference on the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, 2013, pp. 284–286.
- [28] S. Agrawal and P. Khatri, “Facial Expression Detection Techniques: Based on Viola and Jones Algorithm and Principal Component Analysis,” *2015 Fifth International Conference on Advanced Computing & Communication Technologies*, pp. 108–112, 2015.
- [29] M. S. Kalas, “REAL TIME FACE DETECTION AND TRACKING USING OPENCV,” 2014.