Αυτόνομοι Πράκτορες

Εκπαίδευση πράκτορα για αναγνώριση προσώπου με ή χωρίς μάσκα, με χρήση δικτύων Deep learning

Κατσούπης Ευάγγελος, 2017030077

2021

Περιεχόμενα

1 Περιγραφή Προβλήματος

Κατά το ξέσπασμα της πανδημίας SARS-CoV-2 και έχοντας ανακαλύψει τους τρόπους με τους οποίους μεταδίδεται ο ιός, φτάσαμε στο συμπέρασμα, πως η χρήση προστατευτικής μάσκας βοηθά στην μείωση εξάπλωσης αυτού. Πιο συγκεκριμένα, σε κοινόχρηστους χώρους, κλειστούς και μη, όπως πανεπιστήμια, βιβλιοθήκες, θέατρα κτλ., όπου πολλά, άγνωστα μεταξύ τους, άτομα κινούνται σε κοντινές αποστάσεις ή/και χρησιμοποιούν τα ίδια αντικείμενα είναι αναγκαία η χρήση μάσκας.

Για την επίλυση του προβλήματος κατασκευάστηκε ένα συνελικτικό νευρωνικό δίκτυο (Convolutional Neural network ή CNN) το οποίο εκπαιδεύτηκε κατάλληλα σε υπάρχουσα βάση δεδομένων. Ο παραπάνω πράκτορας χρησιμοποιείται για την επεξεργασία εικόνων, οι οποίες προέρχονται από video, και για εξαγωγή αποτελεσμάτων.

Αναλύωνται δύο μοντέλα και ανάλογα με την απόδοσή τους επιλέχθηκε το καλύτερο για την επίλυση του προβλήματος. Παράλληλα, γίνεται εξήγηση των εργαλείων που χρησιμοποιήθηκαν αλλά και η ανάλυση απόδοσης των μοντέλων.

2 Agent

2.1 Εργαλεία που χρησιμοποιήθηκαν

Για την κατασκευή του δικτύου, έγινε χρήση του Keras API. Το Keras είναι ένα API βαθιάς εκμάθησης γραμμένο σε Python, που τρέχει στην πλατφόρμα μηχανικής εκμάθησης TensorFlow.

Το μοντέλο του δικτύου, βασίστηκε στην επεξεργασία δεδομένων μέσο της διαδικασίας Deep Learning, DL. Το DP είναι μια υποκατηγορία της μηχανικής μάθησης που χρησιμοποιεί αλγόριθμους εμπνευσμένους από τη δομή και τη λειτουργία των νευρωνικών δικτύων του εγκεφάλου. Οι αλγόριθμοι βασίζονται στην επεξεργασία και συλλογή δεδομένων με την χρήση πολλαπλών νευρώνων, όπου κάθε ένας από αυτούς λαμβάνει πληροφορία, την επεξεργάζεται και εξάγει ένα αποτέλεσμα. Ποιο συγκεκριμένα πολλαπλασιάζουν κάθε είσοδό τους με το αντίστοιχο συναπτικό βάρος και υπολογίζουν το ολικό άθροισμα των γινομένων. Το άθροισμα αυτό τροφοδοτείται ως όρισμα σε μια συνάρτηση ενεργοποίησης, την οποία υλοποιεί εσωτερικά κάθε κόμβος. Η τιμή που λαμβάνει η συνάρτηση για το εν λόγω όρισμα είναι και η έξοδος του νευρώνα για τις τρέχουσες εισόδους και βάρη.

Στην υλοποίηση αυτή, γίνεται χρήση του Sequential μοντέλου, που παρέχεται από το Keras και δεν είναι τίποτα άλλο από μια γραμμική στοίβα από layers, όπου κάθε layer αποτελείται από νευρώνες. Νευρώνες από διαφορετικά στρώματα συνδέονται μεταξύ τους, και με τα κατάλληλα βάρη, ανάλογα με το είδος του layer που χρησιμοποιείται.

Για την επεξεργασία εικόνων, που απαιτείται για την επίλυση του προβλήματος, χρησιμοποιούνται συνελικτικά (Convolutional layers, Cl) και υπολογιστικά στρώματα (Dense layers, Dl) των οποίων οι νευρώνες επεξεργάζονται την πληροφορία σε κάθε επίπεδο και την περνάνε στο επόμενο.

- Convolutional layers
  Κάθε νεύρωνας ενώς Cl παίρνει σαν είσοδο μια εικόνα, της εφαρμόζει ένα φίλτρο και εξάγει το αποτελέσματα σε νευρώνες του επόμενου στρώματος. Συνολικά, κάθε layer εφαρμόζει μια σειρά από πολλά φίλτρα όπου κάνουν mapping πάνω στην εικόνα και εξάγουν λεπτομέρειες από αυτή. Τέτοιες λεπτομέριες μπορεί να είναι η ανίχνευση άκρων, φωτεινές και μη περιοχές, πληροφορίες όπου βοηθούν τον δίκτυο να μάθει και να λειτουργεί πάνω σε μοτίβα.

- Dense layers
  Αντίστοιχα, στα Dl, κάθε επίπεδο βοηθά στην αλλαγή της διάστασης της πληροφορίας εισόδου, έτσι ώστε το μοντέλο να μπορεί εύκολα να ορίσει τη σχέση μεταξύ των τιμών των δεδομένων στα οποία αυτό λειτουργεί. Τα επίπεδα αυτά χρησιμοποιούνται συνήθως στο τέλος του μοντέλου και λαμβάνουν σαν είσοδο, την έξοδο των Cl.

Ανάλογα με την πληροφορία, οι έξοδοι κάθε layer παίρνει και τα αντίστοιχα βάρη.

2.2 Κατασκευή Μοντέλου

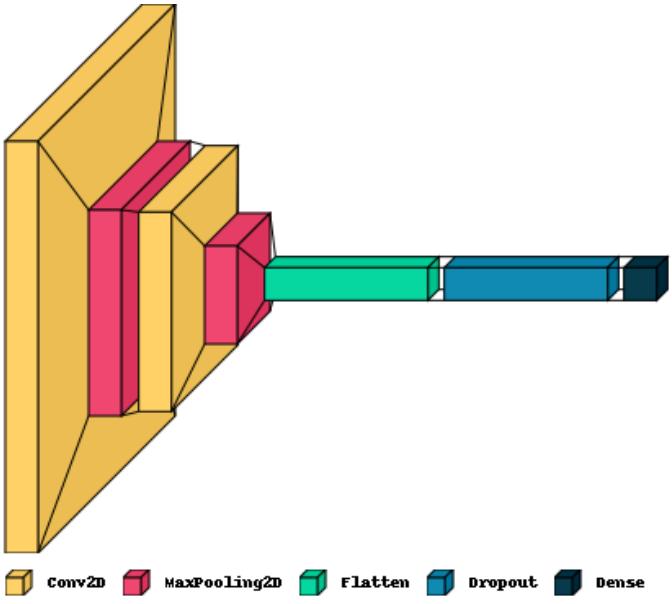
Κατασκευάστηκαν δύο μοντέλα αναφοράς, τα οποία και θα συγκριθούν πιο κάτω.

- Η αρχιτεκτονική του πρώτου μοντέλου αποτελείται από ένα Cl δύο διαστάσεων εισόδου (32 φίλτρων), το οποίο περνά τα δεδομένα σε ένα max-pooling επίπεδο το οποίο εξάγει τα δεδομένα εισόδου, με μειωμένη διάσταση και με μέγιστα χαρακτηριστικά. Το ίδιο μοτίβο ακολουθείται άλλη μια φορά, με την διαφορά στο πλήθος των φίλτρων του Cl ($32 \rightarrow 128$). Τα Cl χρησιμοποιούν την συνάρτηση ενεργοποίησης της Διορθωμένης Γραμμικής Μονάδας (Rectified Linear Unit, relu) για την ενεργοποίηση των αντίστοιχων nodes αλλά και τον υπολογισμό των αντίστοιχων βαρών.

  Αφού μετατρέψουμε τα δεδομένα σε μονοδιάστατη ακολουθία, στην συνέχεια, για να μειωθεί το φαινόμενο του overfitting στα δεδομένα με τα οποία εκπαιδεύεται το μοντέλο (να μην μαθαίνει σε συγκεκριμένα μοτίβα), απενεργοποιείται τυχαία ένα ποσοστό τον νευρώνων που θα μεταφέρουν την πληροφορία στο τελικό layer. Το τελευταίο στάδιο του δικτύου αποτελείται από ένα Dense layer με είσοδο την πληροφορία των τυχαίων ενεργοποιημένων nodes και έξοδο τις δυο πιθανότητες των αντικειμένων, για το οποίο εκπαιδεύτηκε.

  Ποιο συγκεκριμένα το παραπάνω μοντέλο αναλυτικά και γραφικά:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPoolin    (None, 63, 63, 32)        0
 g2D)

 conv2d (Conv2D)             (None, 61, 61, 128)       36992

 max_pooling2d (MaxPoolin    (None, 30, 30, 128)       0
 g2D)

 flatten (Flatten)           (None, 115200)             0

 dropout (Dropout)           (None, 115200)             0

 dense (Dense)               (None, 2)                  230402

=================================================================
Total params: 268,290
Trainable params: 268,290
Non-trainable params: 0
_________________________________________________________________
```

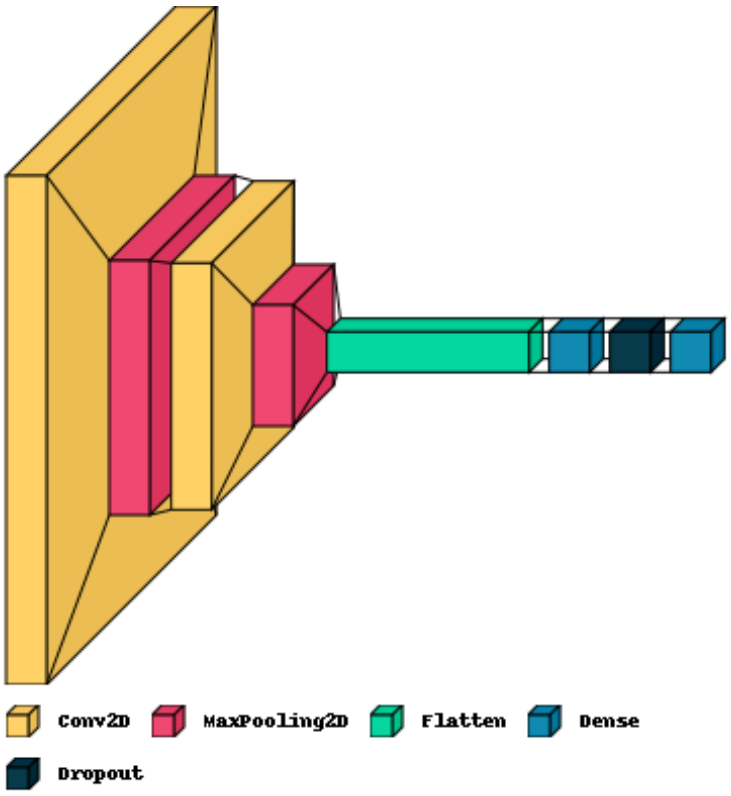


Σχήμα 1: First CNN model

- Αντίστοιχα η αρχιτεκτονική το δεύτερου και τελικού μοντέλου είναι η ίδια με την πρώτη με ένα πρόσθετο Dense layer μεγέθους 64, μετά την μετατροπή σε μονοδιάστατη ακολουθία για να μειωθούν οι διαστάσεις των δεδομένων χωρίς να χαθεί πληροφορία χρησιμοποιώντας υπολογιστική ισχύς (πολλαπλασιασμός πινάκων στους νεύρωνες και υπολογισμός βαρών με την χρήση της συνάρτησης relu).

  Αντίστοιχα Το μοντέλο αυτό αναλυτικά και γραφικά:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPoolin    (None, 63, 63, 32)        0
 g2D)

 conv2d (Conv2D)             (None, 61, 61, 128)       36992

 max_pooling2d (MaxPoolin    (None, 30, 30, 128)       0
 g2D)

 flatten (Flatten)            (None, 115200)            0

 dense (Dense)               (None, 64)                7372864

 dropout (Dropout)            (None, 64)                0

 dense (Dense)               (None, 2)                 130

=================================================================
Total params: 7,410,882
Trainable params: 7,410,882
Non-trainable params: 0
_________________________________________________________________
```



Σχήμα 2: Final CNN model

2.3 Optimizers

Για να εκπαιδευτούν τα μοντέλα εκτός από τα δομικά τους στοιχεία, χρειάζεται να οριστεί ο τρόπος με τον οποίον το δίκτυο θα μαθαίνει καθ'όλη την διάρκεια της εκπαίδευσης. Λόγο, όμως της ύπαρξης του μεγάλου αριθμού παραμέτρων η επιλογή των σωστών βαρών για το μοντέλο και σε γρήγορο, σχετικά, χρόνο αποτελεί ένα δύσκολο πρόβλημα.

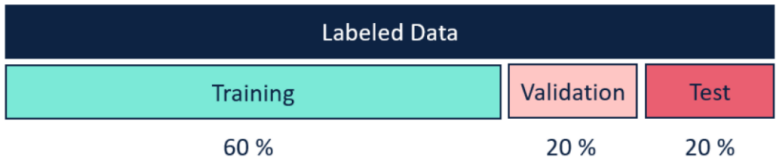
Γι'αυτό τον λόγο γίνετε η χρήση αλγορίθμου βελτίωσης (optimizer) που τροποποιεί τα χαρακτηριστικά του νευρωνικού δικτύου, όπως τα βάρη και τον ρυθμό εκμάθησης, οδηγώντας σε άμεση μείωση συνολικής απώλειας αλλά και βελτιώνοντας την ακρίβεια. Ο optimizer ο οποίος χρησιμοποιείται σε αυτή την περίπτωση είναι ο Adam με ένα χαμηλό, σχετικά learning rate.

3 Data managment

3.1 Διαχωρισμός δεδομένων

Για να εκπαιδευτεί το παραπάνω μοντέλο χρησιμοποιήθηκαν φωτογραφίες που απεικονίζουν ανθρώπους με και χωρίς μάσκα χωρισμένες σε κατηγορίες. Για την ορθή εκπαίδευση αλλά και για να μειωθεί ποιο πολύ το φαινόμενο του overfitting τα κατηγοριοποιημένα δεδομένα (labeled data) χωρίστηκαν σε δεδομένα με τα οποία το μοντέλο εκπαιδεύεται (training) και σε δεδομένα με τα οποία το μοντέλο ελέγχει το πόσο καλά έχει εκπαιδευτεί και τα την διαδικασία της εκμάθησης (validation). Αφού ο πράκτορας έχει εκπαιδευτεί, τέλος, ελέγχεται η αξιοπιστία του, σε ένα τρίτο set δεδομένων, εξάγοντας την πρόβλεψή του για κάθε φωτογραφία αυτού (test).

Ο διαχωρισμός των δεδομένων έγινε στην τύχη και με προϋπόθεση ότι ίδιες φωτογραφίες δεν υπάρχουν σε διαφορετικά set. Το ποσοστό αρχείων σε κάθε set επιλέχθηκε εμπειρικά ώς εξής:



Σχήμα 3: Data splitting

3.2 Προετοιμασία δεδομένων για εκπαίδευση

Για να 'περαστούν' τα δεδομένα σαν είσοδος στο μοντέλο κατά την εκπαίδευση, δημιουργούνται παρτίδες φωτογραφιών από το training set. Λόγο του μικρού πλήθους φωτογραφιών που υπάρχουν διαθέσιμες, από κάθε φωτογραφία δημιουργούνται μορφοποιημένες φωτογραφίες με τυχαίες παραμετροποιήσεις, ώστε να αυξηθεί το διαθέσιμο ωφέλιμο πλήθος.

Την ίδια επεξεργασία δέχονται και τα δεδομένα για το validation όπου και αυτά χωρίζονται σε παρτίδες και περνούν στο δίκτυο.

4 Agent Training

4.1 Δημιουργία data steam

Για να εκπαιδευτεί πιο έγκυρα το δίκτυο, πραγματοποιήθηκε ένας μεγάλος αριθμός εκπαιδεύσεων, όπου κάθε μία από αυτές πρόσφερε στο δίκτυο. Ο αριθμός αυτός (epochs) υποδηλώνει το πόσες φορές θα περάσουν όλα τα δεδομένα από το δίκτυο, δηλαδή θα ολοκληρωθεί ένας κύκλος εκπαίδευσης. Όσο μεγαλύτερος είναι ο αριθμός αυτός, τόσο περισσότερο θα εκπαιδεύονται οι νεύρωνες και άρα τόσο ποιο αξιόπιστα θα είναι τα βάρη.

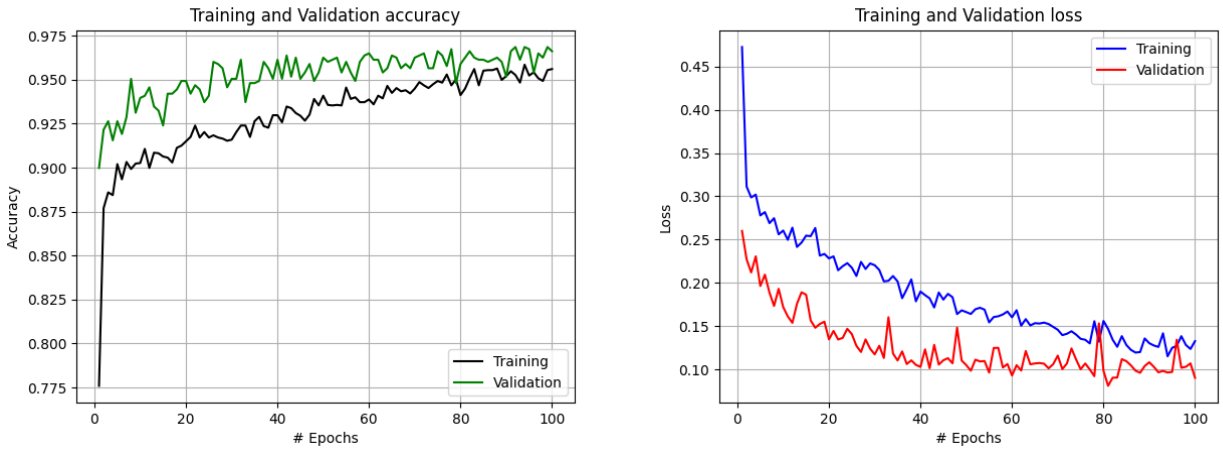
Σε κάθε κύκλο, τα δεδομένα θα εισάγονται στο δίκτυο σε παρτίδες (batches) μεγέθους, κατά κύριο λόγο, μικρότερου του πλήθους των αρχείων προς εκπαίδευση. Γενικά ο τρόπος επιλογής του μεγέθους εξαρτάτε από το πώς θα προσεγγιστεί το πρόβλημα, αλλά και σε γενικούς κανόνες όπως στο ότι ο αριθμός δεν πρέπει να είναι πολύ

μικρός, ώστε να μην αυξάνονται τα σφάλματα, αλλά και ούτε πολύ μεγάλος για ελαχιστοποίηση της απαιτούμενης μνήμης.

4.2 Εκπαίδευση

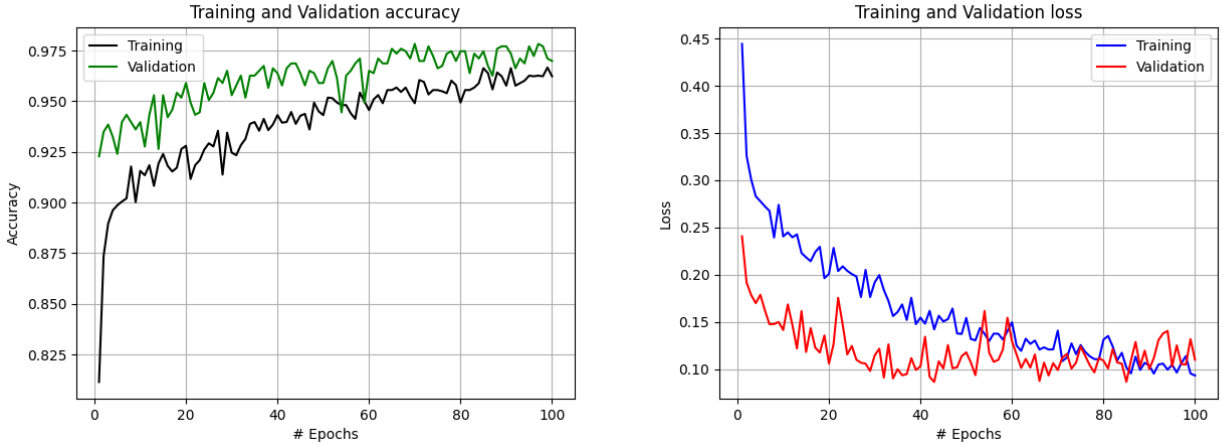
Τα δύο μοντέλα, εκπαιδεύονται για τα ίδια δεδομένα και ελέγχονται η ακρίβεια και η απώλεια τους. Ποιο συγκεκριμένα:

1. Οι χαρακτηριστικές αποτελεσμάτων του πρώτου μοντέλου:



Σχήμα 4: Accuracy-loss plots

2. Αντίστοιχα οι χαρακτηριστικές του δεύτερου:



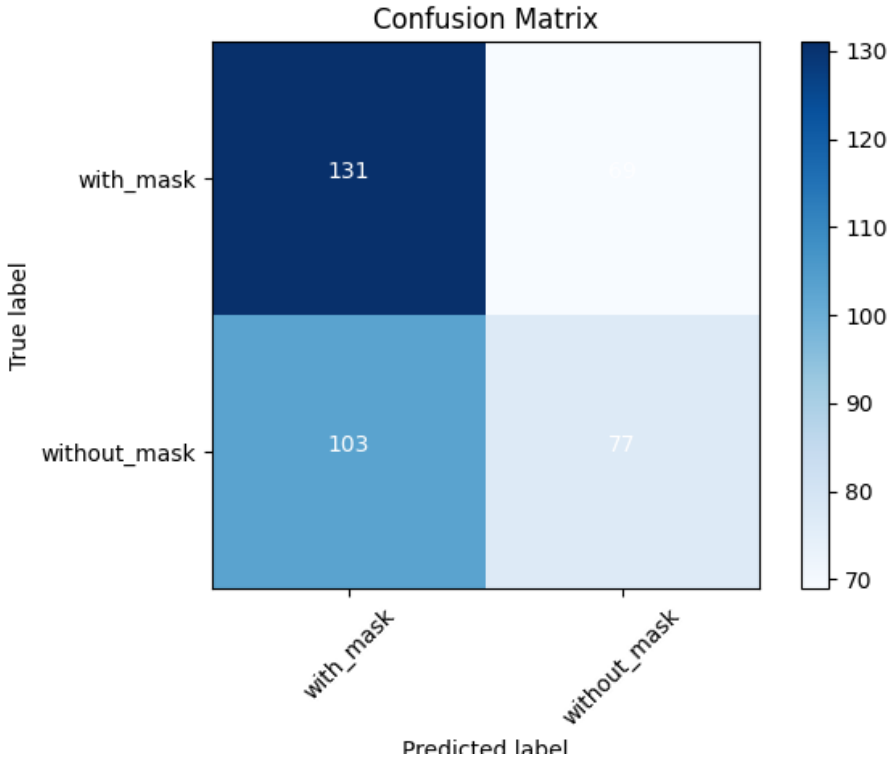
Σχήμα 5: Accuracy-loss plots for model with extra dense layer

Παρατηρείται πως η αρχιτεκτονική με το επιπλέον Dense layer εμφανίζει ελαφρός καλύτερη απόδοση τόσο στην κατά την διάρκεια του training αλλά και του validation, όπως και μικρότερες απώλειες κατά την εκπαίδευση. Χειρότερο φαίνεται να είναι το σφάλμα που εμφανίζει κατά το validation, χωρίς όμως μεγάλη απόκλιση μιας και οι μέσες τιμές και για τα δύο μοντέλα, κατά το στάδιο αυτό, είναι πολύ κοντά. Να μην ξεχαστεί, ότι μιλάμε για αναγνώριση αντικειμένων σε εικόνες και άρα για να επιτευχθούν τιμές, πολύ ποιο κοντά στις ακραίες χρειάζονται παραπάνω δεδομένα εισαγωγής αλλά και 'παραπάνω' χρόνος εκπαίδευσης.

Με βάση τα παραπάνω αποτελέσματα για την επίλυση του προβλήματος χρησιμοποιήθηκε το δεύτερο μοντέλο.

5 Agent Check

Για το έλεγχο του παραπάνω πράκτορα, χρησιμοποιήθηκαν αρχεία από το test set και με την χρήση ενός Confusion Matrix παρουσιάζετε το συνολικό αποτελέσματα. Ποιο συγκεκριμένα κάθε φωτογραφία, που γνωρίζουμε σε ποια κλάση ανήκει, περνά από το δίκτυο και υπολογίζεται η πρόβλεψη της κάθε κλάσης. Για το παράδειγμα:



Σχήμα 6: Confusion Matrix for the test set

Παρατηρείται πως ο πράκτορας δεν τα πήγε και πολύ καλά μιας μεγάλο ποσοστό ανθρώπων χωρίς μάσκα τους αναγνώρισε ώς ανθρώπους με μάσκα. Αντίθετα προέβλεψε την μη ύπαρξη μάσκας σε δείγματα με μάσκα σε πολύ μικρότερο ποσοστό. Αυτό το σχετικά χαμηλό accuracy ($\simeq 55\%$) οφείλεται τόσο στην απλότητα του δικτύου όσου και στην κακή επεξεργασία των εικόνων ελέγχου.

6 Face-mask detection

6.1 Εφαρμογή

Για την επίλυση του αρχικού προβλήματος χρησιμοποιήθηκε παραλλαγή έτοιμης υλοποίησης η οποία, με χρήση κάμερας, συνεχόμενα λαμβάνει εικόνα, αναγνωρίζει που βρίσκεται το πρόσωπο σε αυτή και την στέλνει στον πράκτορα για να πραγματοποιήσει την πρόβλεψη. Για την υλοποίηση αυτή γίνεται χρήση του Object Detection αλγορίθμου, Haar Cascade. Σε περίπτωση που ο χρήστης δεν φορά μάσκα εμφανίζεται κόκκινο περίβλημα γύρο από το πρόσωπό του και σε αντίθετη περίπτωση ένα πράσινο. Για κάθε μία από τις παραπάνω περιπτώσεις εμφανίζεται και το ποσοστό της βεβαιότητας το οποίο υπολόγισε ο πράκτορας.

6.2 Προβλήματα

Λόγο της φτωχής βάσης δεδομένων ο πράκτορας μπορεί να αναγνωρίσει κατά κύριο λόγο μάσκες χειρουργικού τύπου και αρκετές σκουρόχρωμες ίδιου τύπου ή/και αυξημένης προστασίας.

7 References

[Fundamentals] Deep Learning Fundamentals - Classic Edition (link).

[Agent creation and training] TensorFlow - Python Deep Learning Neural Network API (link)

[Face Mask Detector] Face Mask Detector with Python (link)

[Data managment] TUC ECE TEL311 (link)

[Data sets] Images (link), (link)