

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΣΤΟΝ ΙΣΤΟ ΤΩΝ ΠΡΑΓΜΑΤΩΝ

Θεοδώρα Αναστασίου ΑΜ: itp19103

Άγγελος Λίχας ΑΜ: itp19123

Κώστας Μητσοκάπας ΑΜ: itp19124

Περιεχόμενα

[Προετοιμασία των δεδομένων 2](#_Toc38835295)

[Δεδομένα 3](#_Toc38835296)

[Μοντέλο 4](#_Toc38835297)

# Προετοιμασία των δεδομένων

Αρχικά πήραμε τα δεδομένα και τα σπάσαμε σε frames για το training, το validation και το testing. Για το frame του training πήραμε το 70% των δεδομένων, για το validation το 20% και για το testing 10%.

Στα δεδομένα μας αναλύοντας την στήλη findings χωρίσαμε τις εικόνες των ασθενών σε 2 φακέλους, covid σε αυτούς που έχουν κορωνοϊό και other σε αυτούς που δεν έχουν.

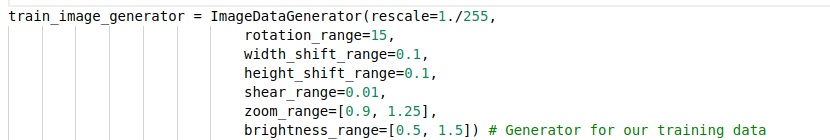
Έπειτα χωρίσαμε ξανά σε φακέλους train, valid, test για το κάθε frame και μέσα τους covid και other για να έχουμε τα labels για να μπορέσουμε να εξετάσουμε τις κλάσεις ξεχωριστά.



# Δεδομένα

Για την διαχείριση των δεδομένων μας, δηλαδή τις εικόνες των ασθενών, χρησιμοποιήσαμε το imageDataGenerator για κάθε σετ ξεχωριστά. Οι παράμετροι που επιλέξαμε είναι οι εξής:

* Rescale: ώστε να μικρύνουμε τις φωτογραφίες
* Rotation\_range: για να προβλέψουμε τυχαίες περιστροφές
* Shear\_range: εύρος διακύμανσης
* Zoom\_range: πρόβλεψη zoom στις εικόνες
* Horizontal\_flip: οριζόντια περιστροφή
* Vertical\_flip: κάθετη περιστροφή
* Fill\_mode: μέθοδος για την συμπλήρωση
* Data\_format: πως θα είναι τα δεδομένα της φωτογραφίας
* Brightness\_range: το εύρος της φωτεινότητας των εικόνων

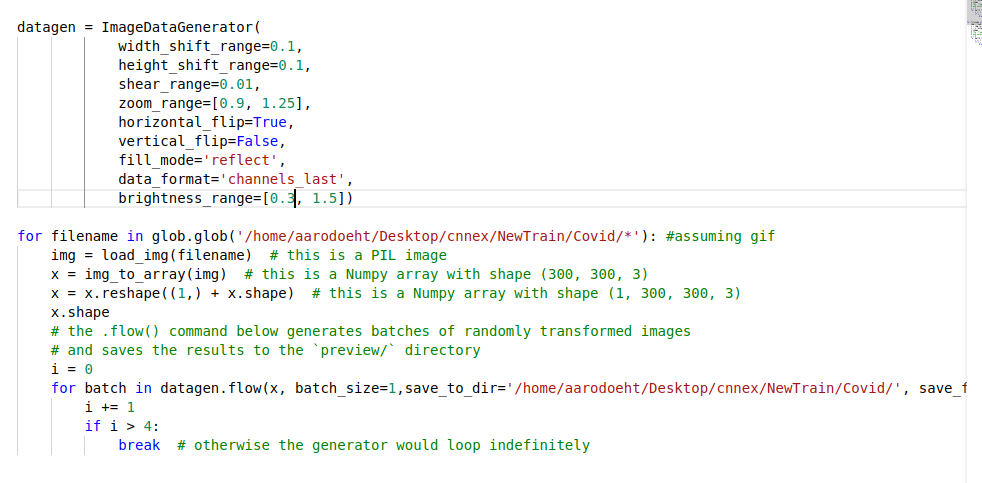


Έπειτα με την χρήση του flow\_from\_directory διαβάζουμε τις εικόνες από τους υποφακέλους και τα χωρίζει σε batches από τις augmented εικόνες και ορίζουμε το class mode και το target size. Στην περίπτωση μας ορίσαμε ως class mode binary και target size 350x350.

Με την ίδια διαδικασία διαχειριστήκαμε και τα δεδομένα του validation και του test με λιγότερη επεξεργασία των εικόνων στο imageDataGenerator.

# Μοντέλο

Λόγω της έλλειψης αρκετών δεδομένων δημιουργήσαμε έξτρα εικόνες(περίπου 3-4 για κάθε μία από τις παλιές), ώστε να παράγουμε παραπάνω δεδομένα και να εκπαιδευτεί το μοντέλο καλύτερα.



Ορίζουμε ως:

STEP\_SIZE\_TRAIN = train\_generator.n//train\_generator.batch\_size

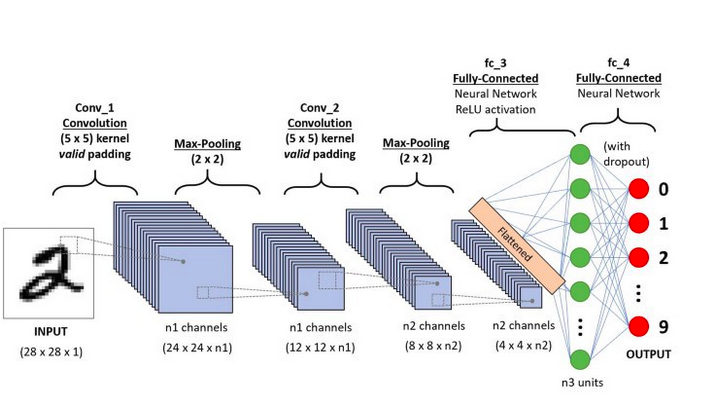
Για τα βήματα του training διαιρούμε το πλήθος των δεδομένων του train με το batch size τους.

STEP\_SIZE\_VALID = validation\_generator.n//validation\_generator.batch\_size

Για τα βήματα του validation διαιρούμε το πλήθος των δεδομένων του train με το batch size τους.

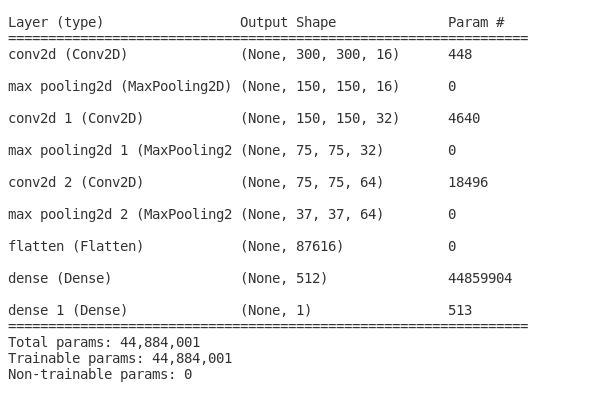
STEP\_SIZE\_TEST = test\_generator.n//test\_generator.batch\_size

Για τα βήματα του test διαιρούμε το πλήθος των δεδομένων του train με το batch size τους.



Για το νευρωνικό ορίσαμε:

3 convolutional επίπεδα και 3 max-pooling επίπεδα, έπειτα έχουμε ένα flatten επίπεδο που ακολουθείτε από 2 dense επίπεδα με ένα dropout μεταξύ τους



Οι optimizers που χρησιμοποιήσαμε και δοκιμάσαμε είναι:

Adam, sgd, RMSprop κλπ.

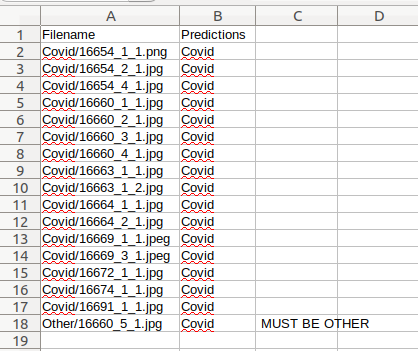
Αυτός που καταλήξαμε είναι ο sgd λόγω των αποτελεσμάτων και με learning rate 1e-6

Οι εποχές που δοκιμάσαμε ήταν από 2 έως 10 αλλά καταλήξαμε στις 10.

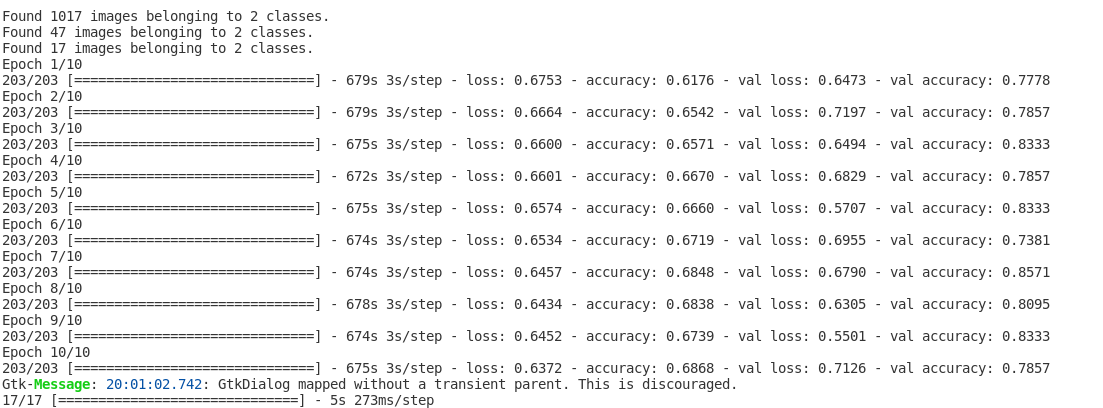
Τα batches που ορίσαμε είναι:

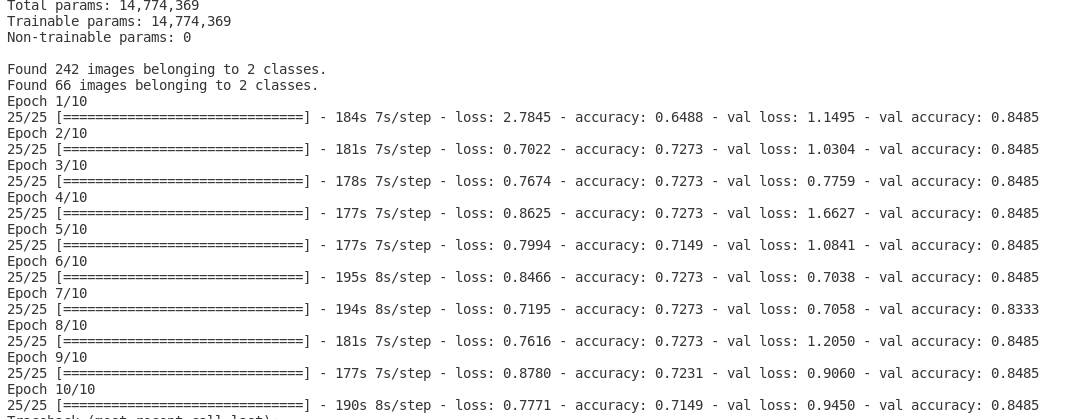
* Training: 10 batches
* Validation: 5 batches
* Test: 1 batch

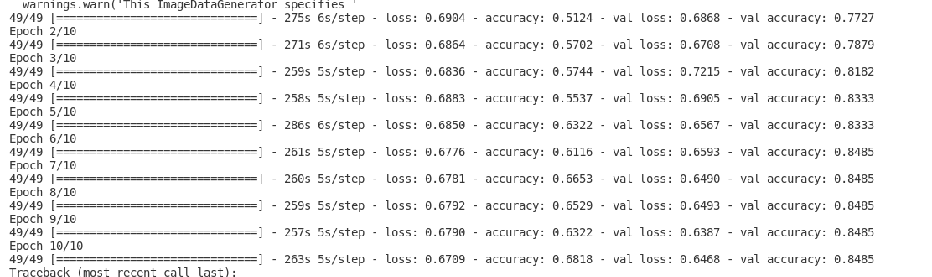
## Prediction

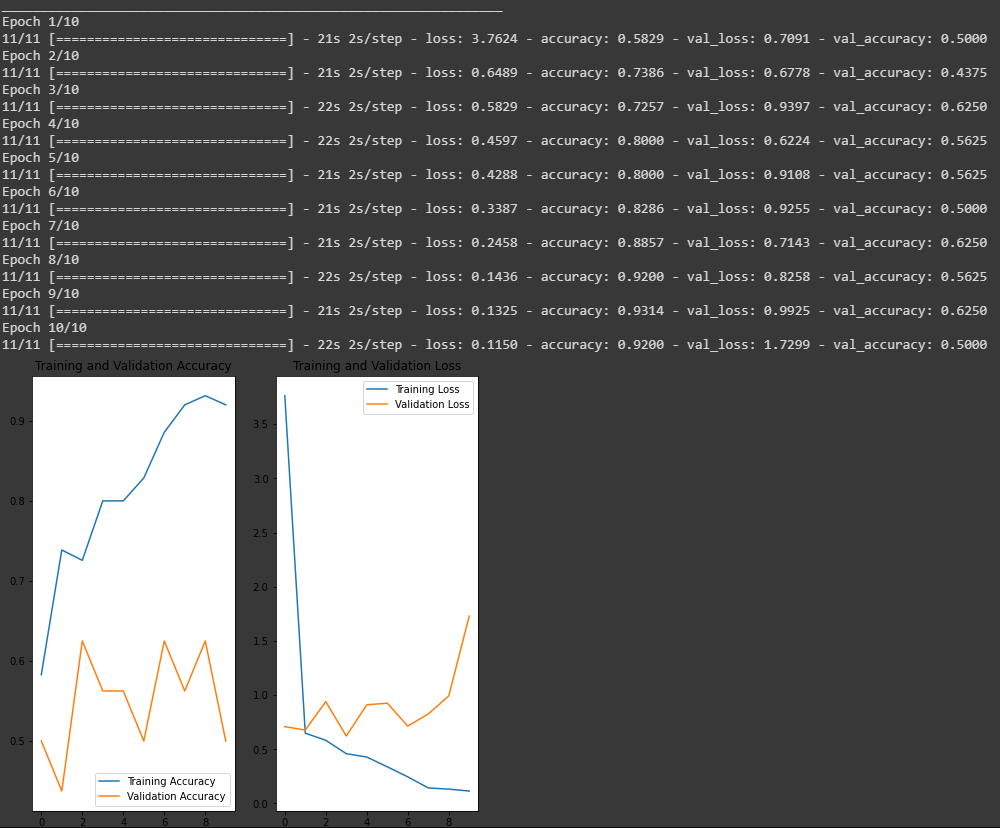
Δημιουργήσαμε προβλέψεις για το μοντέλο μας, λαμβάνοντας τις πιθανότητες του ασθενούς να έχει κορωνοϊό. Έπειτα αυτές τις αποθηκεύει σε ένα csv αρχείο. ( Παραθέτουμε το αρχείο results.csv)  


## Δοκιμές









## Γραφικές Παραστάσεις

