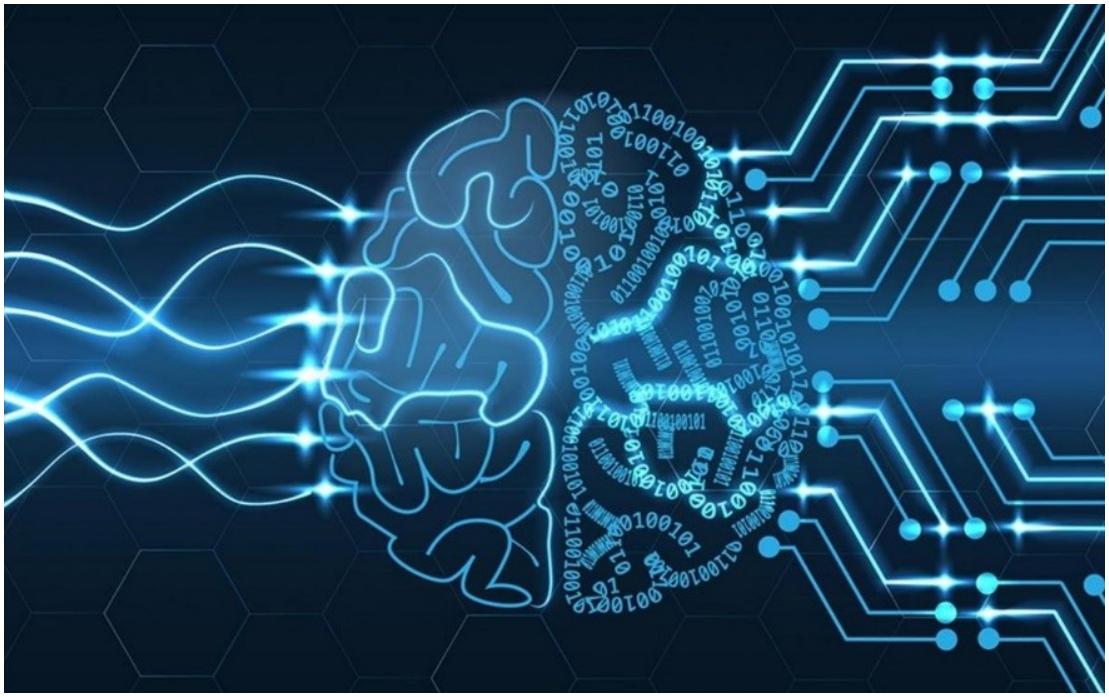


# ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΣΤΟΝ ΙΣΤΟ ΤΩΝ ΠΡΑΓΜΑΤΩΝ

Θεοδώρα Αναστασίου ΑΜ: itp19103
Άγγελος Λίχας ΑΜ: itp19123
Κώστας Μητσοκάπας ΑΜ: itp19124

# Περιεχόμενα

# 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.

```
Found 1353 images belonging to 2 classes.
Found 47 images belonging to 2 classes.
Found 17 images belonging to 2 classes.
```

# 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

```
train_image_generator = ImageDataGenerator(rescale=1./255,
                                            rotation_range=15,
                                            width_shift_range=0.1,
                                            height_shift_range=0.1,
                                            shear_range=0.01,
                                            zoom_range=[0.9, 1.25],
                                            brightness_range=[0.5, 1.5]) # Generator for our training data
```

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.

```
datagen = ImageDataGenerator(
            width_shift_range=0.1,
            height_shift_range=0.1,
            shear_range=0.01,
            zoom_range=[0.9, 1.25],
            horizontal_flip=True,
            vertical_flip=False,
            fill_mode='reflect',
            data_format='channels_last',
            brightness_range=[0.3, 1.5])

for filename in glob.glob('/home/aarodoeht/Desktop/cnnex/NewTrain/Covid/*'): #assuming gif
    img = load_img(filename)  # this is a PIL image
    x = img_to_array(img)  # this is a Numpy array with shape (300, 300, 3)
    x = x.reshape((1,) + x.shape)  # this is a Numpy array with shape (1, 300, 300, 3)
    x.shape
    # the .flow() command below generates batches of randomly transformed images
    # and saves the results to the `preview/` directory
    i = 0
    for batch in datagen.flow(x, batch_size=1,save_to_dir='/home/aarodoeht/Desktop/cnnex/NewTrain/Covid/', save_f
        i += 1
        if i > 4:
            break  # otherwise the generator would loop indefinitely
```

Ορίζουμε ως:

STEP_SIZE_TRAIN = train_generator.n//train_generator.batch_size

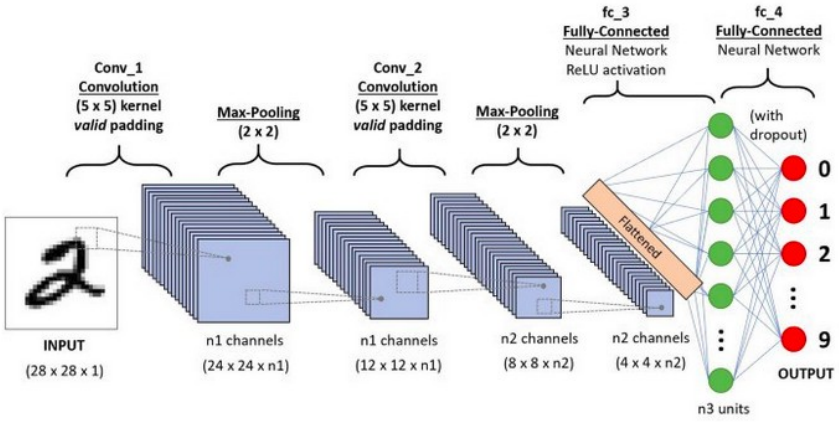
Για τα βήματα του training διαιρούμε το πλήθος των δεδομένων του train με το batch size τους.

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

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 300, 300, 16)      448

max pooling2d (MaxPooling2D) (None, 150, 150, 16)      0

conv2d 1 (Conv2D)            (None, 150, 150, 32)      4640

max pooling2d 1 (MaxPooling2 (None, 75, 75, 32)        0

conv2d 2 (Conv2D)            (None, 75, 75, 64)        18496

max pooling2d 2 (MaxPooling2 (None, 37, 37, 64)        0

flatten (Flatten)            (None, 87616)             0

dense (Dense)                (None, 512)               44859904

dense 1 (Dense)              (None, 1)                 513
=================================================================
Total params: 44,884,001
Trainable params: 44,884,001
Non-trainable params: 0
```

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)

| | A | B | C | D |
|---|---|---|---|---|
| 1 | Filename | Predictions | | |
| 2 | Covid/16654_1_1.png | Covid | | |
| 3 | Covid/16654_2_1.jpg | Covid | | |
| 4 | Covid/16654_4_1.jpg | Covid | | |
| 5 | Covid/16660_1_1.jpg | Covid | | |
| 6 | Covid/16660_2_1.jpg | Covid | | |
| 7 | Covid/16660_3_1.jpg | Covid | | |
| 8 | Covid/16660_4_1.jpg | Covid | | |
| 9 | Covid/16663_1_1.jpg | Covid | | |
| 10 | Covid/16663_1_2.jpg | Covid | | |
| 11 | Covid/16664_1_1.jpg | Covid | | |
| 12 | Covid/16664_2_1.jpg | Covid | | |
| 13 | Covid/16669_1_1.jpeg | Covid | | |
| 14 | Covid/16669_3_1.jpeg | Covid | | |
| 15 | Covid/16672_1_1.jpg | Covid | | |
| 16 | Covid/16674_1_1.jpg | Covid | | |
| 17 | Covid/16691_1_1.jpg | Covid | | |
| 18 | Other/16660_5_1.jpg | Covid | MUST BE OTHER | |
| 19 | | | | |

# ΔΟΚΙΜΈΣ

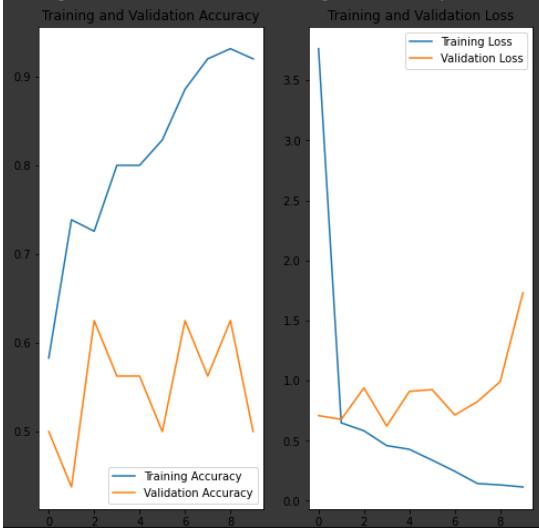
```
Found 1017 images belonging to 2 classes.
Found 47 images belonging to 2 classes.
Found 17 images belonging to 2 classes.
Epoch 1/10
203/203 [==============================] - 679s 3s/step - loss: 0.6753 - accuracy: 0.6176 - val loss: 0.6473 - val accuracy: 0.7778
Epoch 2/10
203/203 [==============================] - 679s 3s/step - loss: 0.6664 - accuracy: 0.6542 - val loss: 0.7197 - val accuracy: 0.7857
Epoch 3/10
203/203 [==============================] - 675s 3s/step - loss: 0.6600 - accuracy: 0.6571 - val loss: 0.6494 - val accuracy: 0.8333
Epoch 4/10
203/203 [==============================] - 672s 3s/step - loss: 0.6601 - accuracy: 0.6670 - val loss: 0.6829 - val accuracy: 0.7857
Epoch 5/10
203/203 [==============================] - 675s 3s/step - loss: 0.6574 - accuracy: 0.6660 - val loss: 0.5707 - val accuracy: 0.8333
Epoch 6/10
203/203 [==============================] - 674s 3s/step - loss: 0.6534 - accuracy: 0.6719 - val loss: 0.6955 - val accuracy: 0.7381
Epoch 7/10
203/203 [==============================] - 674s 3s/step - loss: 0.6457 - accuracy: 0.6848 - val loss: 0.6790 - val accuracy: 0.8571
Epoch 8/10
203/203 [==============================] - 678s 3s/step - loss: 0.6434 - accuracy: 0.6838 - val loss: 0.6305 - val accuracy: 0.8095
Epoch 9/10
203/203 [==============================] - 674s 3s/step - loss: 0.6452 - accuracy: 0.6739 - val loss: 0.5501 - val accuracy: 0.8333
Epoch 10/10
203/203 [==============================] - 675s 3s/step - loss: 0.6372 - accuracy: 0.6868 - val loss: 0.7126 - val accuracy: 0.7857
Gtk-Message: 20:01:02.742: GtkDialog mapped without a transient parent. This is discouraged.
17/17 [==============================] - 5s 273ms/step
```

```
Total params: 14,774,369
Trainable params: 14,774,369
Non-trainable params: 0
```

```
Found 242 images belonging to 2 classes.
Found 66 images belonging to 2 classes.
Epoch 1/10
25/25 [==============================] - 184s 7s/step - loss: 2.7845 - accuracy: 0.6488 - val loss: 1.1495 - val accuracy: 0.8485
Epoch 2/10
25/25 [==============================] - 181s 7s/step - loss: 0.7022 - accuracy: 0.7273 - val loss: 1.0304 - val accuracy: 0.8485
Epoch 3/10
25/25 [==============================] - 178s 7s/step - loss: 0.7674 - accuracy: 0.7273 - val loss: 0.7759 - val accuracy: 0.8485
Epoch 4/10
25/25 [==============================] - 177s 7s/step - loss: 0.8625 - accuracy: 0.7273 - val loss: 1.6627 - val accuracy: 0.8485
Epoch 5/10
25/25 [==============================] - 177s 7s/step - loss: 0.7994 - accuracy: 0.7149 - val loss: 1.0841 - val accuracy: 0.8485
Epoch 6/10
25/25 [==============================] - 195s 8s/step - loss: 0.8466 - accuracy: 0.7273 - val loss: 0.7038 - val accuracy: 0.8485
Epoch 7/10
25/25 [==============================] - 194s 8s/step - loss: 0.7195 - accuracy: 0.7273 - val loss: 0.7058 - val accuracy: 0.8333
Epoch 8/10
25/25 [==============================] - 181s 7s/step - loss: 0.7616 - accuracy: 0.7273 - val loss: 1.2050 - val accuracy: 0.8485
Epoch 9/10
25/25 [==============================] - 177s 7s/step - loss: 0.8780 - accuracy: 0.7231 - val loss: 0.9060 - val accuracy: 0.8485
Epoch 10/10
25/25 [==============================] - 190s 8s/step - loss: 0.7771 - accuracy: 0.7149 - val loss: 0.9450 - val accuracy: 0.8485
```

```
  warnings.warn('This ImageDataGenerator specifies '
49/49 [==============================] - 275s 6s/step - loss: 0.6904 - accuracy: 0.5124 - val loss: 0.6868 - val accuracy: 0.7727
Epoch 2/10
49/49 [==============================] - 271s 6s/step - loss: 0.6854 - accuracy: 0.5702 - val loss: 0.6708 - val accuracy: 0.7879
Epoch 3/10
49/49 [==============================] - 259s 5s/step - loss: 0.6836 - accuracy: 0.5744 - val loss: 0.7215 - val accuracy: 0.8182
Epoch 4/10
49/49 [==============================] - 258s 5s/step - loss: 0.6883 - accuracy: 0.5537 - val loss: 0.6965 - val accuracy: 0.8333
Epoch 5/10
49/49 [==============================] - 286s 6s/step - loss: 0.6850 - accuracy: 0.6322 - val loss: 0.6567 - val accuracy: 0.8333
Epoch 6/10
49/49 [==============================] - 261s 5s/step - loss: 0.6776 - accuracy: 0.6116 - val loss: 0.6593 - val accuracy: 0.8485
Epoch 7/10
49/49 [==============================] - 260s 5s/step - loss: 0.6781 - accuracy: 0.6653 - val loss: 0.6490 - val accuracy: 0.8485
Epoch 8/10
49/49 [==============================] - 259s 5s/step - loss: 0.6792 - accuracy: 0.6529 - val loss: 0.6493 - val accuracy: 0.8485
Epoch 9/10
49/49 [==============================] - 257s 5s/step - loss: 0.6790 - accuracy: 0.6322 - val loss: 0.6387 - val accuracy: 0.8485
Epoch 10/10
49/49 [==============================] - 263s 5s/step - loss: 0.6789 - accuracy: 0.6818 - val loss: 0.6468 - val accuracy: 0.8485
Traceback (most recent call last):
```

```
Epoch 1/10
11/11 [==============================] - 21s 2s/step - loss: 3.7624 - accuracy: 0.5829 - val_loss: 0.7091 - val_accuracy: 0.5000
Epoch 2/10
11/11 [==============================] - 21s 2s/step - loss: 0.6489 - accuracy: 0.7386 - val_loss: 0.6778 - val_accuracy: 0.4375
Epoch 3/10
11/11 [==============================] - 22s 2s/step - loss: 0.5829 - accuracy: 0.7257 - val_loss: 0.9397 - val_accuracy: 0.6250
Epoch 4/10
11/11 [==============================] - 22s 2s/step - loss: 0.4597 - accuracy: 0.8000 - val_loss: 0.6224 - val_accuracy: 0.5625
Epoch 5/10
11/11 [==============================] - 21s 2s/step - loss: 0.4288 - accuracy: 0.8000 - val_loss: 0.9108 - val_accuracy: 0.5625
Epoch 6/10
11/11 [==============================] - 21s 2s/step - loss: 0.3387 - accuracy: 0.8286 - val_loss: 0.9255 - val_accuracy: 0.5000
Epoch 7/10
11/11 [==============================] - 21s 2s/step - loss: 0.2458 - accuracy: 0.8857 - val_loss: 0.7143 - val_accuracy: 0.6250
Epoch 8/10
11/11 [==============================] - 22s 2s/step - loss: 0.1436 - accuracy: 0.9200 - val_loss: 0.8258 - val_accuracy: 0.5625
Epoch 9/10
11/11 [==============================] - 21s 2s/step - loss: 0.1325 - accuracy: 0.9314 - val_loss: 0.9925 - val_accuracy: 0.6250
Epoch 10/10
11/11 [==============================] - 22s 2s/step - loss: 0.1150 - accuracy: 0.9200 - val_loss: 1.7299 - val_accuracy: 0.5000
```



## ΓΡΑΦΙΚΈΣ ΠΑΡΑΣΤΆΣΕΙΣ