Deep Learning Exam SoftUni 2018

Radoslav Ivanov

Classifications of Images (Iconography vs Painting)

I have decided to use <https://www.kaggle.com/thedownhill/art-images-drawings-painting-sculpture-engraving> dataset, but only two categories from this dataset (Iconography vs Painting). I have training set with 2077 iconographies images and 2042 paintings images, for validation set I have 231 iconographies images and 228 paintings images.

For testing I used 187 images from google. Iconographies (95) (<https://www.google.bg/search?biw=1229&bih=603&tbm=isch&sa=1&ei=bj4IXLvBIcuEwPAPiayB4A8&q=iconography+painting+gpeg&oq=iconography+painting+gpeg&gs_l=img.3...76819.82963..83144...3.0..0.116.777.6j2......1....1..gws-wiz-img.......0i30.hIM7RQrPqg4> ) Paintings (92) (<https://www.google.com/search?biw=1536&bih=706&tbm=isch&sa=1&ei=JT4RXNmuKerJrgTcwJegCQ&q=painting+museum+portraits&oq=painting+museum+portraits&gs_l=img.3...0.0..893908...0.0..0.0.0.......1......gws-wiz-img.FB3s79LJVR0> ).

**Dataset**: <https://drive.google.com/open?id=1Pd3rXJG6Hb9kuFKf7q5o1XRdhl3qWBDr>

**Google account**: 4/sgC2qJ3OWT-poq\_F0J1HJSS76ET7ywdFWFGtLDMQ0sV3GpHFY7FPung

**Notebook file**: <https://drive.google.com/open?id=1-aWr-cibZPyGah7BviY01tGR8Ti6nN1P>

**Weights**: <https://drive.google.com/open?id=1qcLq_r1_3akmktsWT8H5YlKCnHa5Y0_x>

If you want to run model with already calculated weights (without training), you can put **TRAINED\_MODEL = True**.

My goal is to make classification for these images, which return Iconography or Painting.

I used CNN with the following architecture:

Conv2D(32, (3, 3), padding='same, activation='relu')

Conv2D(32, (3, 3), padding='same', activation='relu')

MaxPool2D(pool\_size=(2, 2))

Conv2D(64, (3, 3), padding='same', activation='relu')

Conv2D(64, (3, 3), padding='same', activation='relu')

MaxPool2D(pool\_size=(2, 2))

Conv2D(128, (3, 3), padding='same', activation='relu')

Conv2D(128, (3, 3), padding='same', activation='relu')

MaxPool2D(pool\_size=(2, 2))

Conv2D(256, (3, 3), padding='same', activation='relu')

Conv2D(256, (3, 3), padding='same', activation='relu')

MaxPool2D(pool\_size=(2, 2))

Flatten()

Dense(256, activation='relu')

Dropout(0.5)

Dense(256, activation='relu')

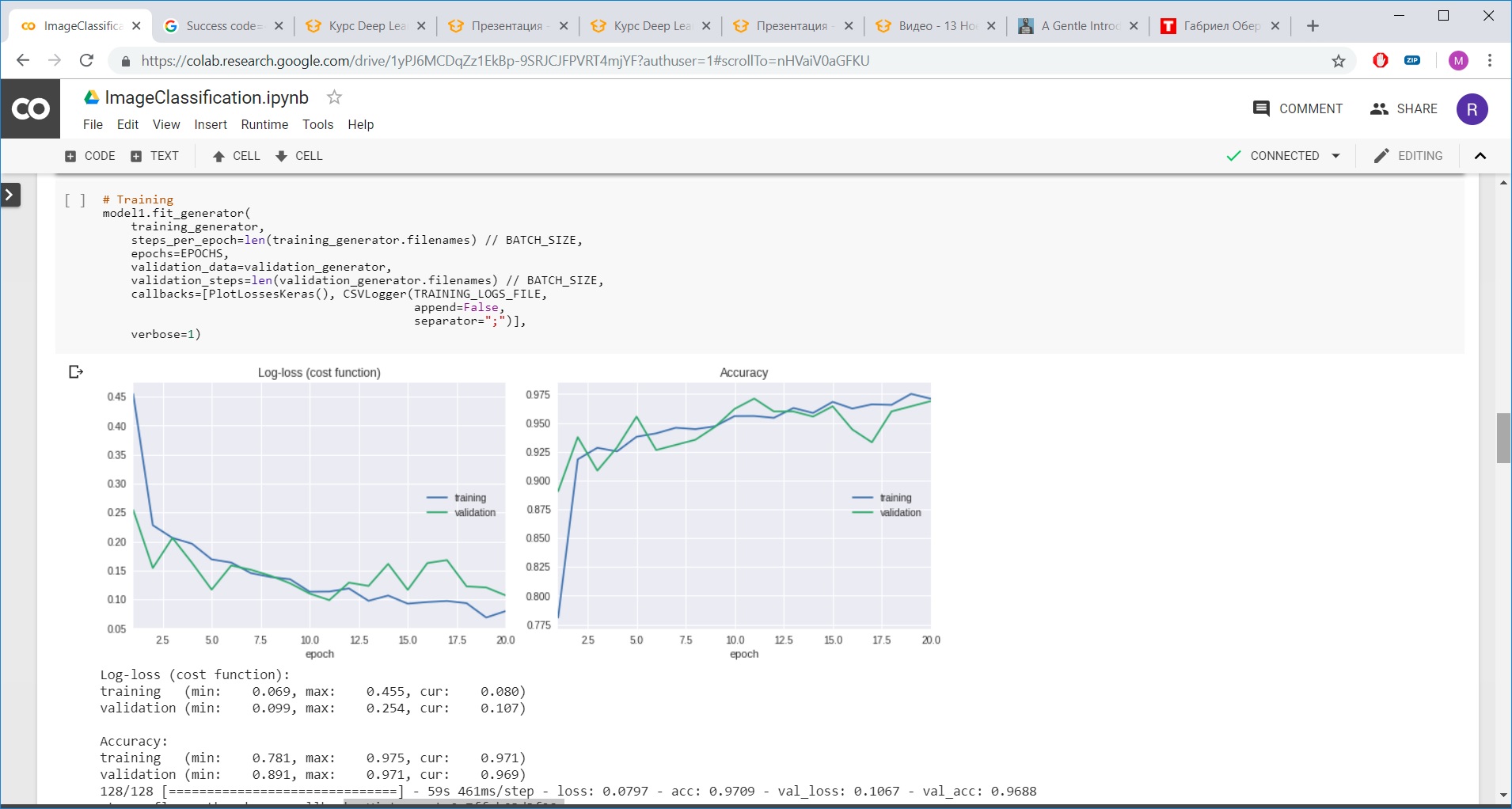
Dropout(0.5)

Dense(1, activation='sigmoid')

For loss function I choose binary cross entropy, for optimizer AdamOptimizer(with different learning rate) and accuracy for metric.

## **Model 1** Learning rate = 0.0001

## Max Accuracy 0.975 for training and validation 0.971

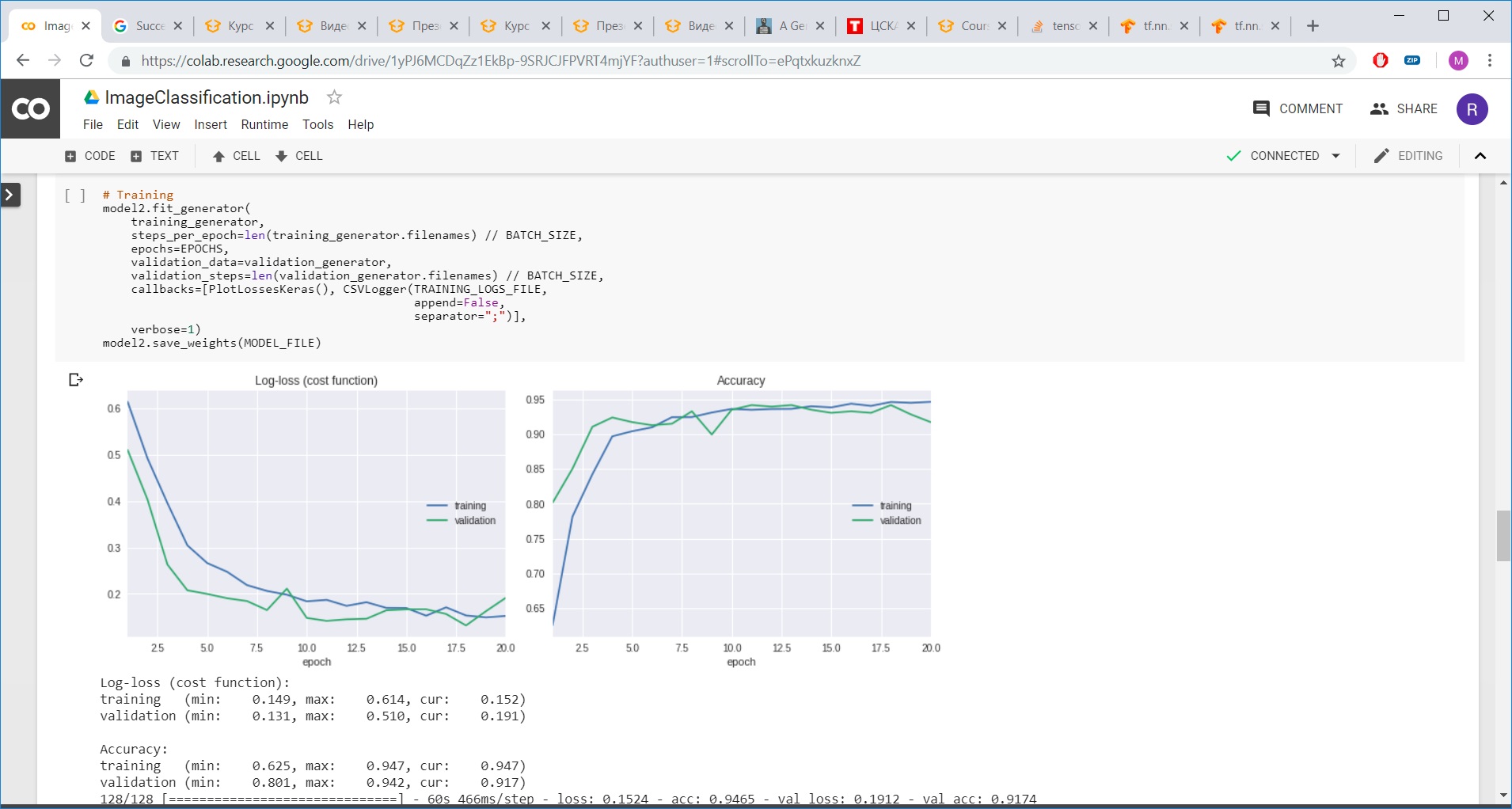


Let’s assumed that the "real" error in dataset is 0.5% and we have:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Train set error** | **Validation set error** | **Bias, %** | **Variance, %** | **Verdict** |
| Model1 | 3% | 3% | 2,5% | 0% | **Bias** |

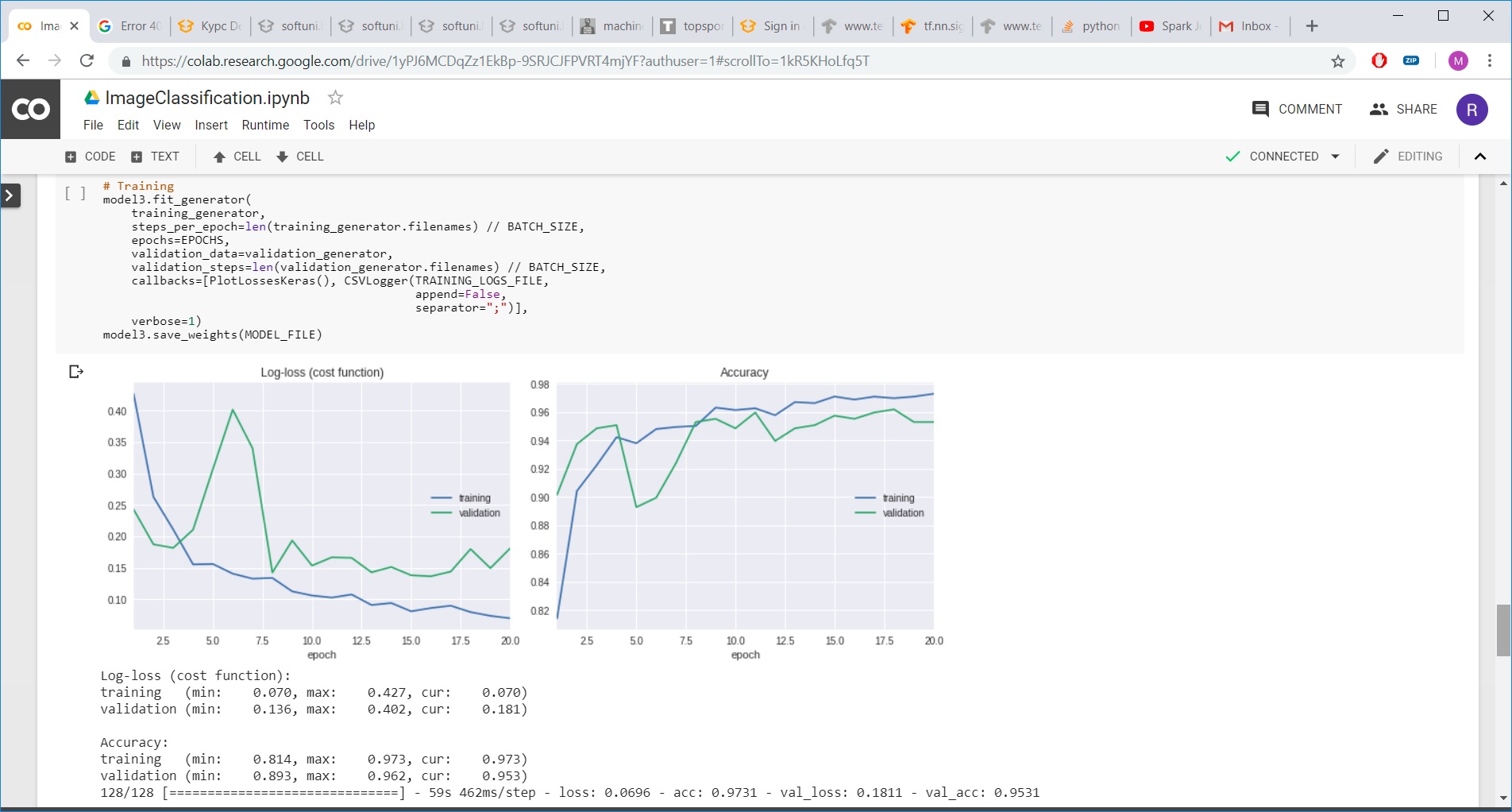
## **Model 2** Learning rate = 0.00001

## Max Accuracy 0.947 for training and validation 0.942



## **Model 3** Learning rate = 0.001

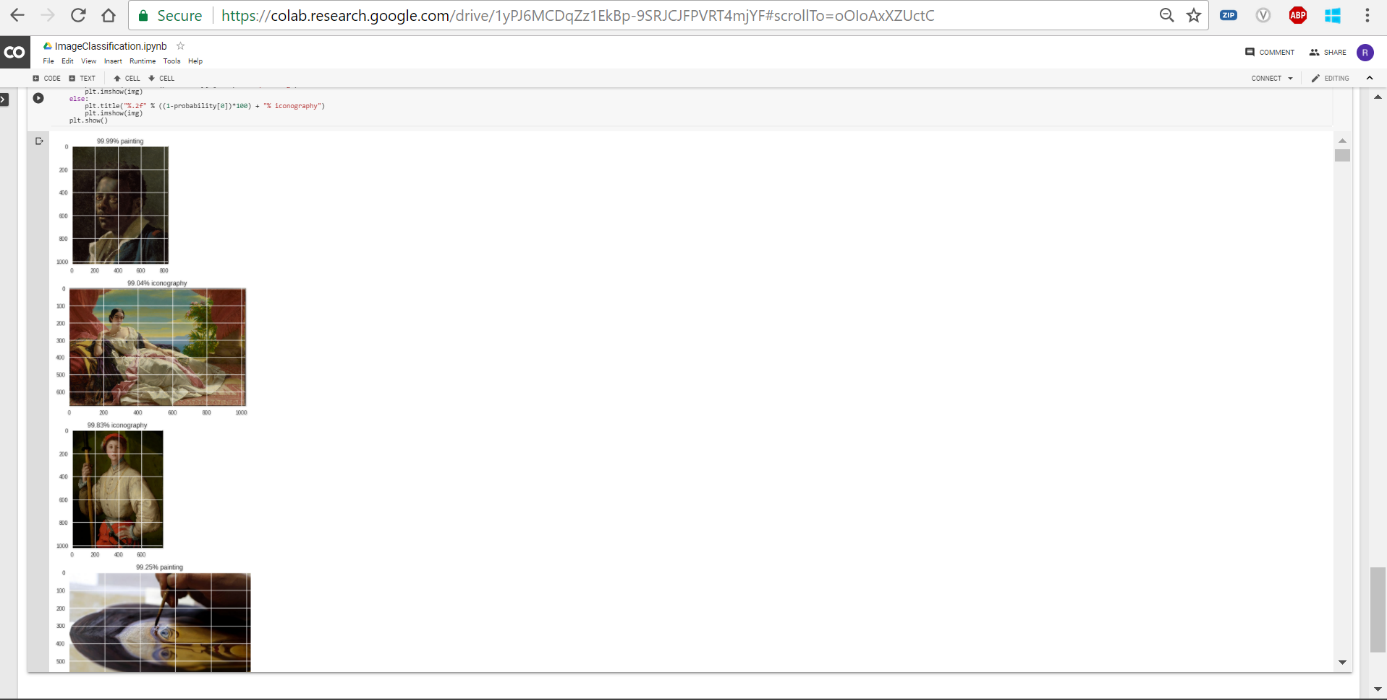
## Max Accuracy 0.973 for training and validation 0.962



# Image Classification:

## Conclusion

With this CNN we achieve very good results in dataset (around 97%), because dataset is structured and we have selected images. In real life, images from google we are far away from this result.



# Reference:

<https://softuni.bg/trainings/1921/deep-learning-november-2018>

<https://medium.com/@parthvadhadiya424/hello-world-program-in-keras-with-cnn-dog-vs-cat-classification-efc6f0da3cc5>