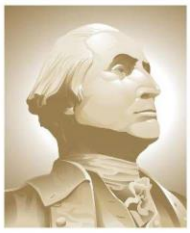


THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

FACE MASK

DETECTION

Report by: Osemekhian Ehilen

OVERVIEW

The importance of face masks[1] has become mandatory to dampen the spread of COVID-19 virus through the human respiratory channels.

This project will show and compare machine learning models on facemask prediction. The project was carried out by two people; and my part was to leverage some pre-trained models in predicting a facemask dataset of images to show one of the three classes- 'with mask', 'no mask' or 'not wearing mask correctly'. While the other participant built a basic Convolutional Neural Network model to carry out this task.

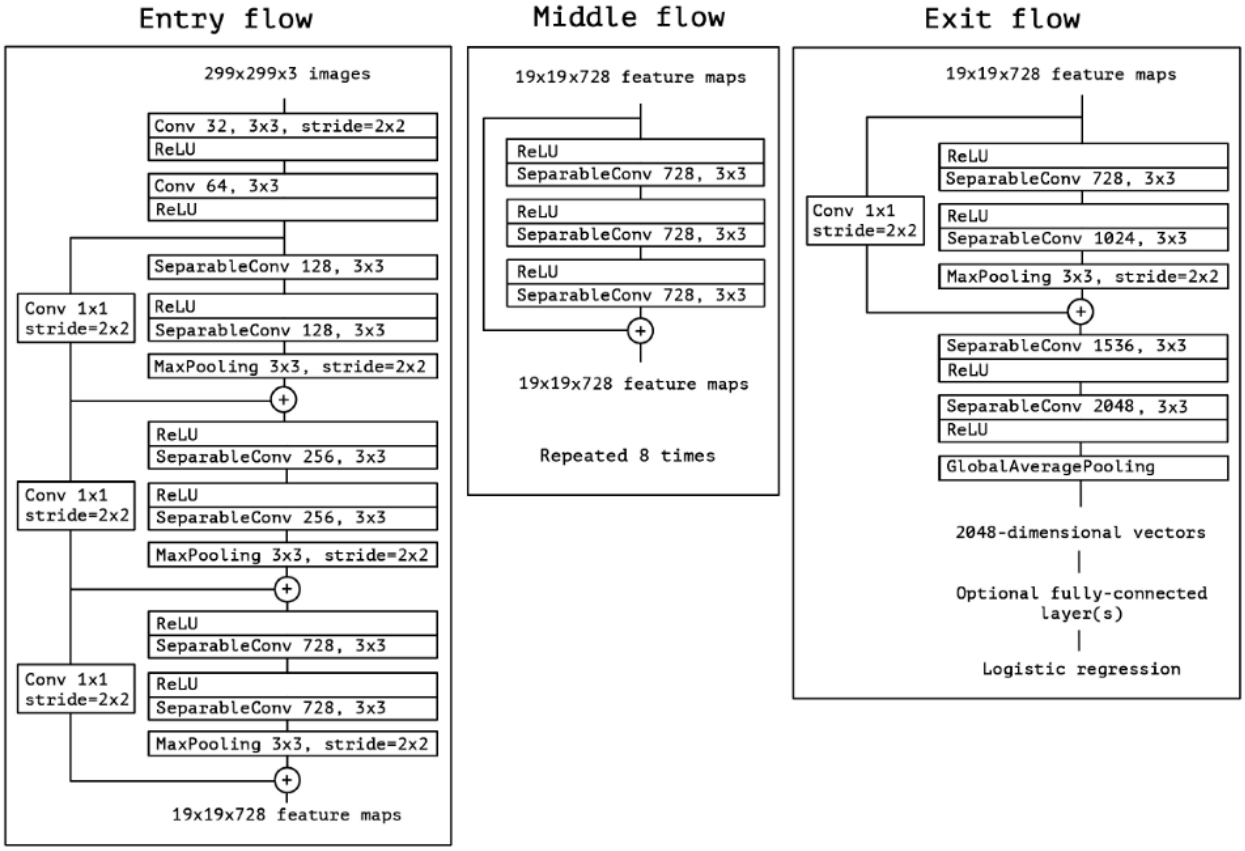
Note that I reduced the classes to two because the samples for not wearing mask correctly looked like they wore the mask correctly.

[1] Centers for Disease Control and Prevention (5 April 2020). "What to Do if You Are Sick". U.S. Centers for Disease Control and Prevention (CDC). Archived from the original on 14 February 2020. Retrieved 24 April 2020.

WORK DESCRIPTION

With transfer learning helping us to leverage on pre-trained model to solve other similar problems, I took into consideration three which are:

- Xception: is an extreme version of Inception with a modified depthwise separable convolution which was first runner up in ILSVRC [2]2015



Source: https://arxiv.org/pdf/1610.02357.pdf

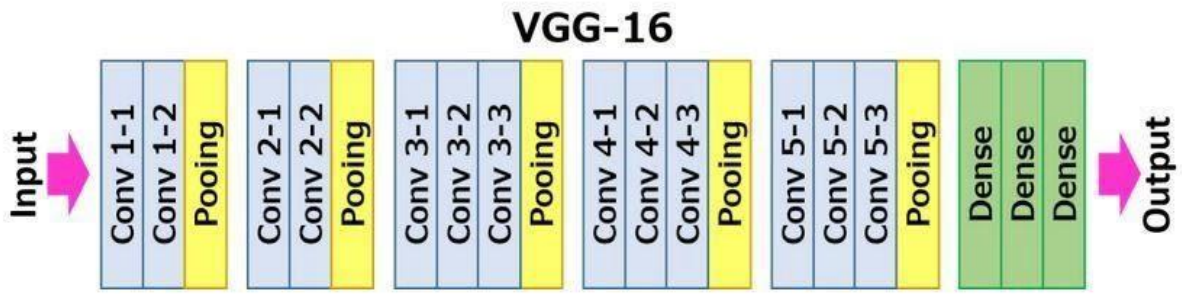
- ResNet50:

[2] Xception- With Depthwise Separable Convolution, Better Than Inception-V3(Image Classifcation), https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Source: https://iq.opengenus.org/resnet50-architecture/

- VGG16: VGG16 was created for object detection and classification algorithm and it is popular amongst top transfer learning algorithms.
  Rohini G[3] said the 16 in VGG16 refers to 16 layers that have weights. In VGG16 there are thirteen convolutional layers, five Max Pooling layers, and three Dense layers which sum up to 21 layers but it has only sixteen weight layers i.e., learnable parameters layer.



Source: https://miro.medium.com/max/828/0*6VP81rFoLWp10FcG

[3] Everything You Need To Know About VGG16, https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918#:~:text=with%20transfer%20learning.-,VGG16%20Architecture,layers%20i.e.%2C%20learnable%20parameters%20layer.

Data Pre-Processing

During pre-processing, a function that parses the images and annotations from its folders to a data frame was used and thanks to jiaowoguanren[4].

Then, a split into train, validation and test set were achieved from the sci-kit learn package in python. TensorFlow's Image Generation function was then used to parse to images and labels into tensors, thereby rescaling (dividing the pixels in images by 255), flipping, shifting, zooming, rotation was performed.

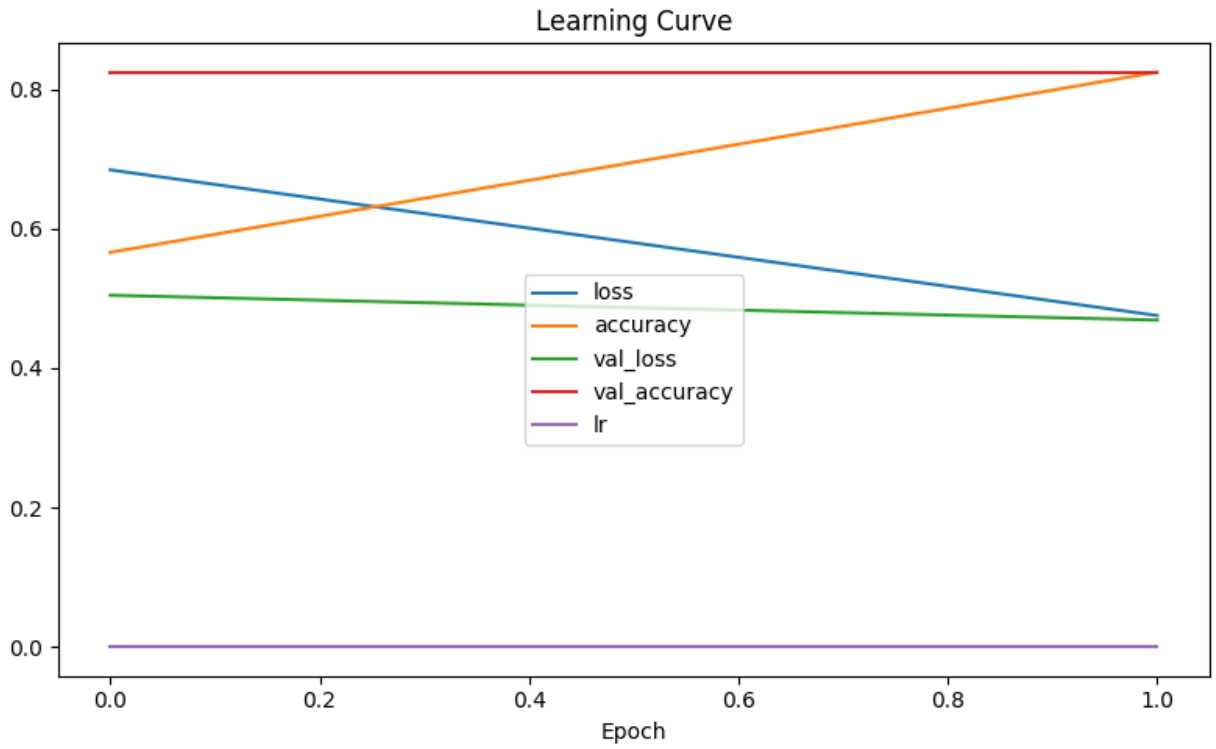
Also, the default pre-processing function for each pre-trained model were applied to the images during their own training.

Training and Results

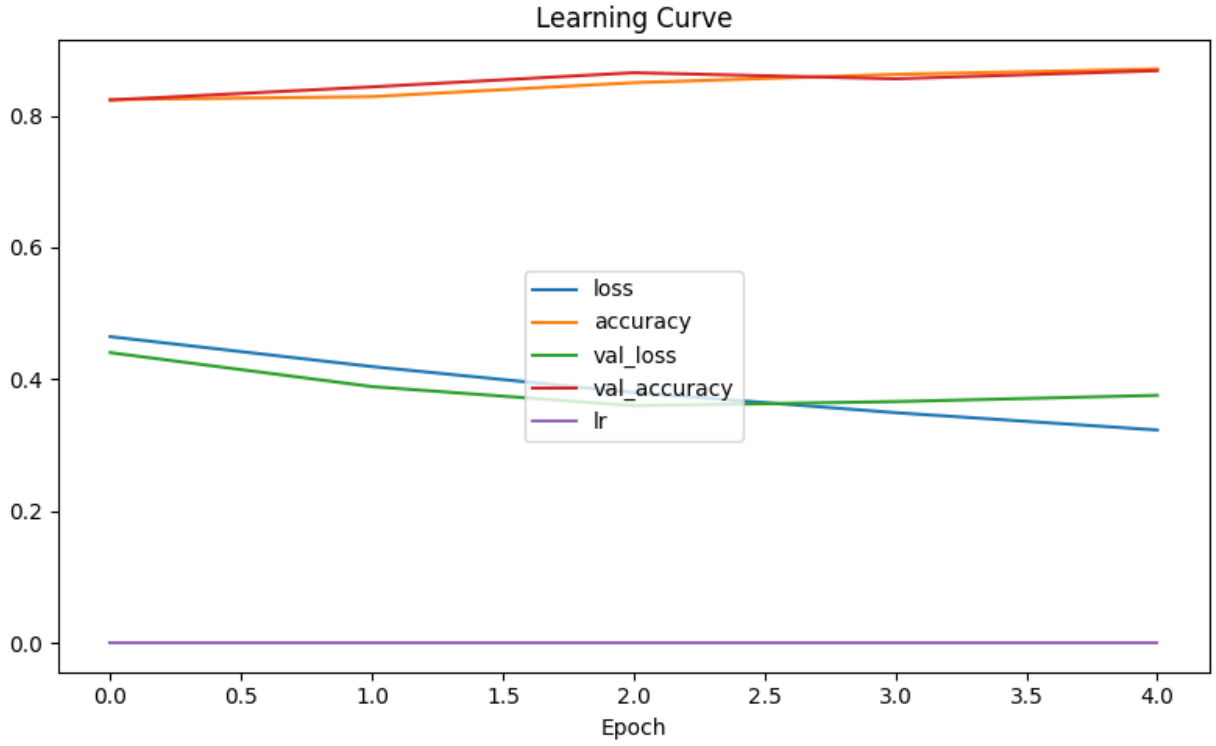
A global average pooling layer and a fully connected or dense layer with number of class output were added to the pre-trained models.

First, I froze all layers of the pre-trained models and trained it to see its performance. Observing VGG16 learning curve which was similar in Xception and ResNet50:

[4] https://www.kaggle.com/code/jiaowoguanren/face-mask-detection-tensorflow-cnn-resmlp

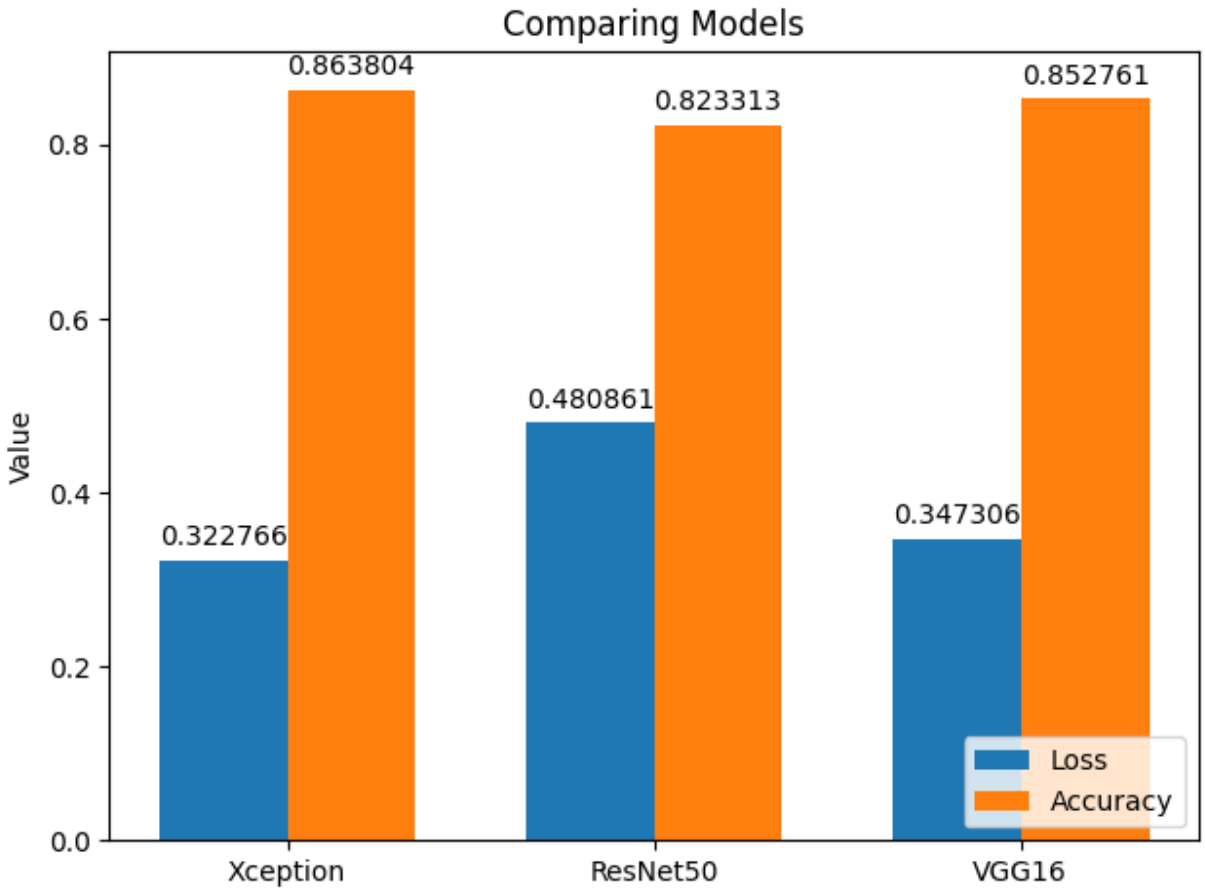


Secondly, I unfroze the layers to be trainable and the learning curve which was similar in the three cases of pre-trained models:



After testing the model on the test set and validation set, I discovered that the "mask not worn correctly" class were misclassified always and the number of images for that class were greatly outnumbered by "with mask" and "without mask". I decided to change that outnumbered class to without mask class because I want the aim is for the model to get those without masks.

After training I compared the three models as mentioned earlier on the test set:

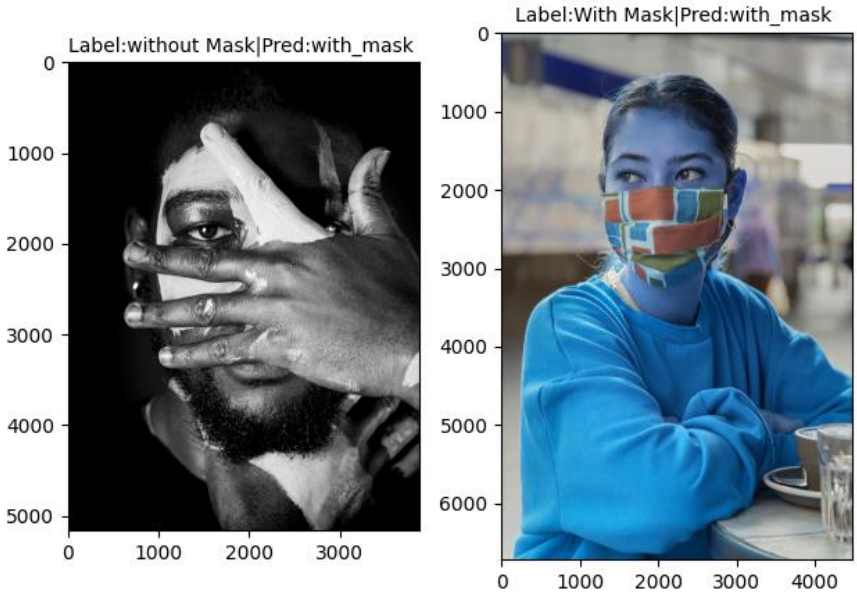


Xception performed best followed by VGG16 with ResNet5o performing least for this dataset.

Testing with Best Model (Xception)

I went further in testing with my image and images from unsplash [5]website.

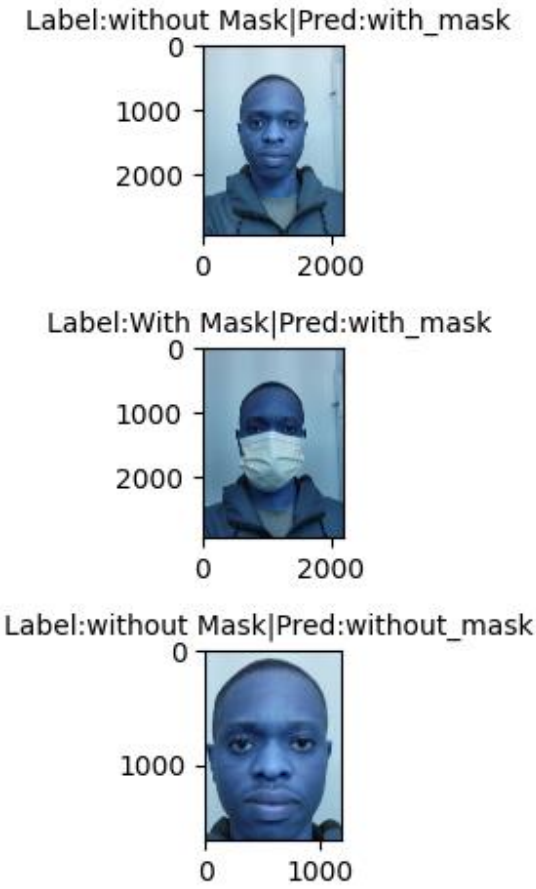
Here the images are correctly classified.



[5] https://unsplash.com/s/photos/facemask

The model seems to think that the person's hand is the facemask on the left image but correctly classified the image on the right.

I discovered that the model behaves well when the person's face is detected/cropped out as you can see below.



Summary and Conclusion

The Xception model amongst ResNet50 and VGG16 performed best for this dataset.

Though with 86% accuracy the model is sensitive to the face alone to predict correctly and confidently.

During the course of this project, I have learnt:

- How to parse images into tensorflow understandable format using its image generation function.
- How to use pre-trained models for prediction.
- How to pre-processing images.
- How to leverage cloud platform like AWS to boost my training power with cloud-based GPU.

Code from internet with 43 lines copied and 3 lines edited with 278 of mine giving:

$\frac{43-3}{15+278} * 100 = 13.7\%$ of code from internet.

Recommendation

I would suggest using an object detection model to hypothesize object's location as this would help in giving great confidence in predicting people with or without facemasks as our model suffered from the object's location.

A **Faster R-CNN** can be leveraged for this course since its good at detecting objects.

References

https://vijayabhaskar96.medium.com/tutorial-image-classification-with-keras-flow-from-directory-and-generators-95f75ebe5720

https://arxiv.org/abs/1506.01497