**Face Mask Detector with deep learning**

* Classify images of human with mask or not by using **transfer learning** from a pre-trained network.
* Customize a pretrained model

1. **Feature Extraction**, Use the representations learned by a previous network to extract meaningful features from new samples.
2. **Fine-Tuning**, Unfreeze a few of the top layers of a frozen model base and jointly train both the newly-added classifier layers and the last layers of the base model.

* General machine learning workflow:

1. Examine and understand the **dataset**
2. Build an input pipeline, do **augmentation**
3. Compose the **model**

* Load in the pretrained base model (and pretrained weights)
* Stack the classification layers on top

1. **Train** and **Evaluate** model

-----------------------------------------------------------------------------------------------------------------

Transfer learning and fine-tuning: <https://www.tensorflow.org/tutorials/images/transfer_learning>

Github, Face-Mask-Detection: <https://github.com/chandrikadeb7/Face-Mask-Detection>

-----------------------------------------------------------------------------------------------------------------

**STEP1. Train face mask detect model**

1. **Dataset**

* With mask: 1000 images
* Without mask: 514 images

Collect from Bing Search API, Kaggle datasets, RMFD dataset.

Training dataset: Testing dataset = 80%:20% = 4:1 = 1211:303

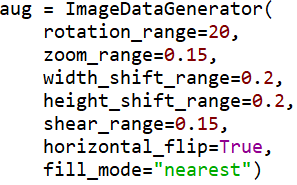
-----------------------------------------------------------------------------------------------------------------



-----------------------------------------------------------------------------------------------------------------

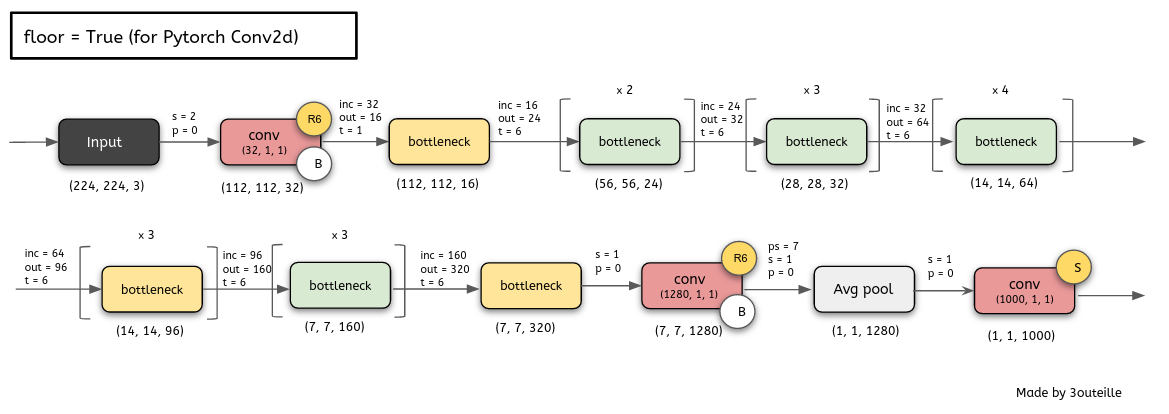
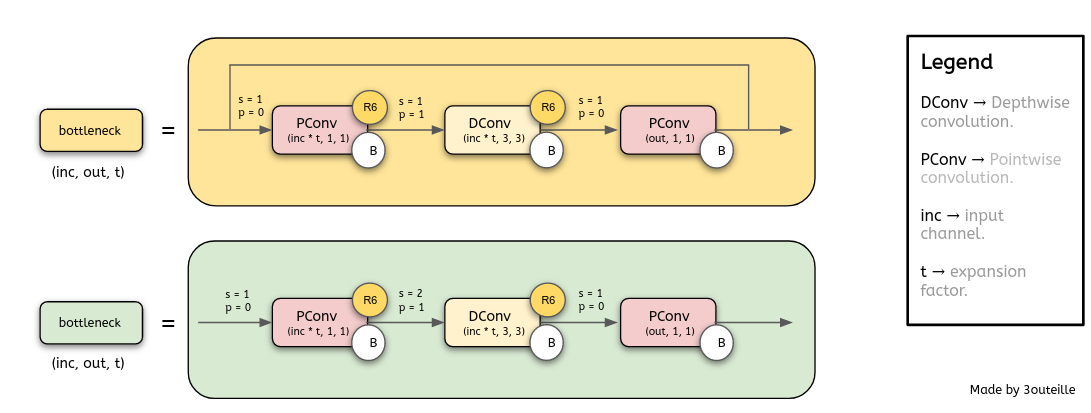
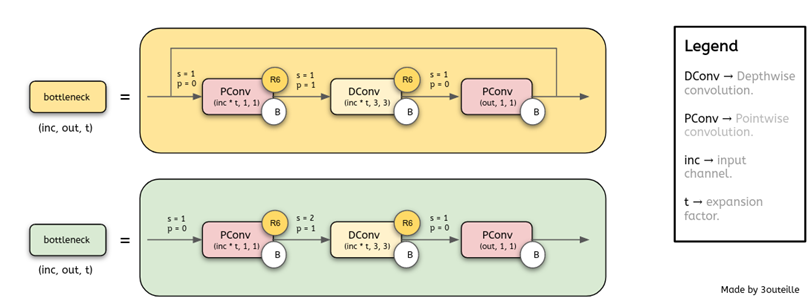
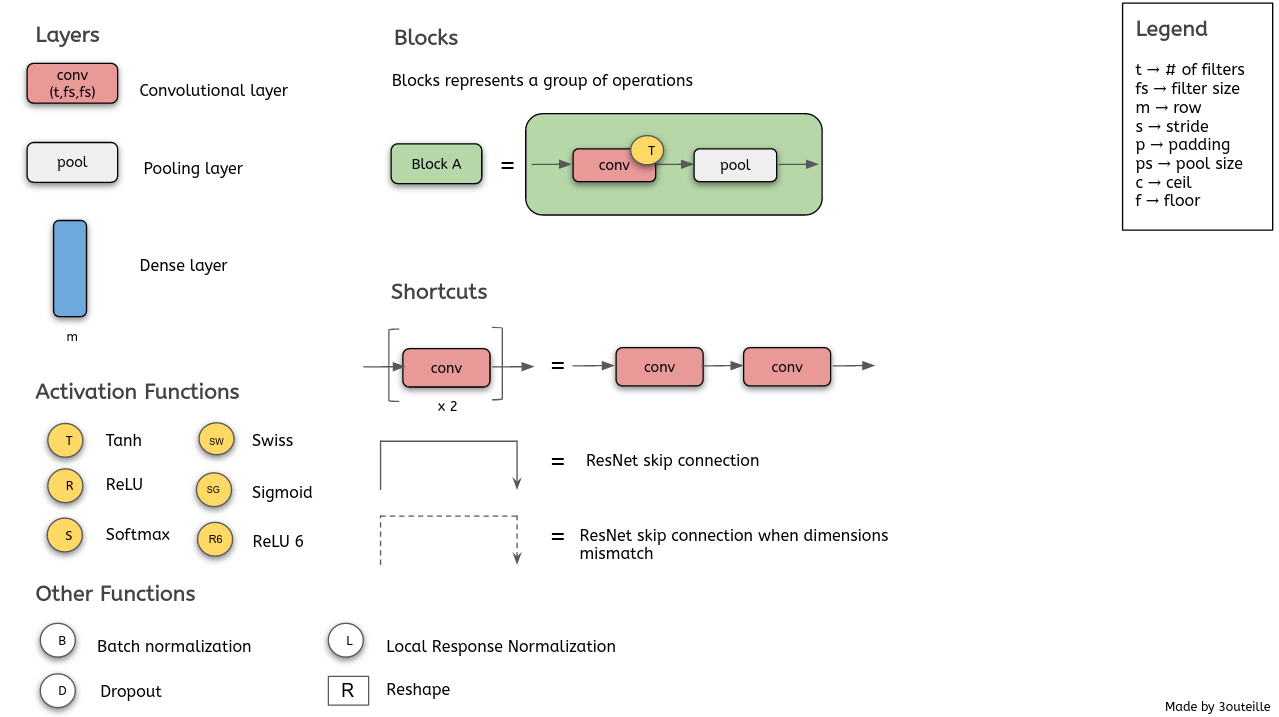
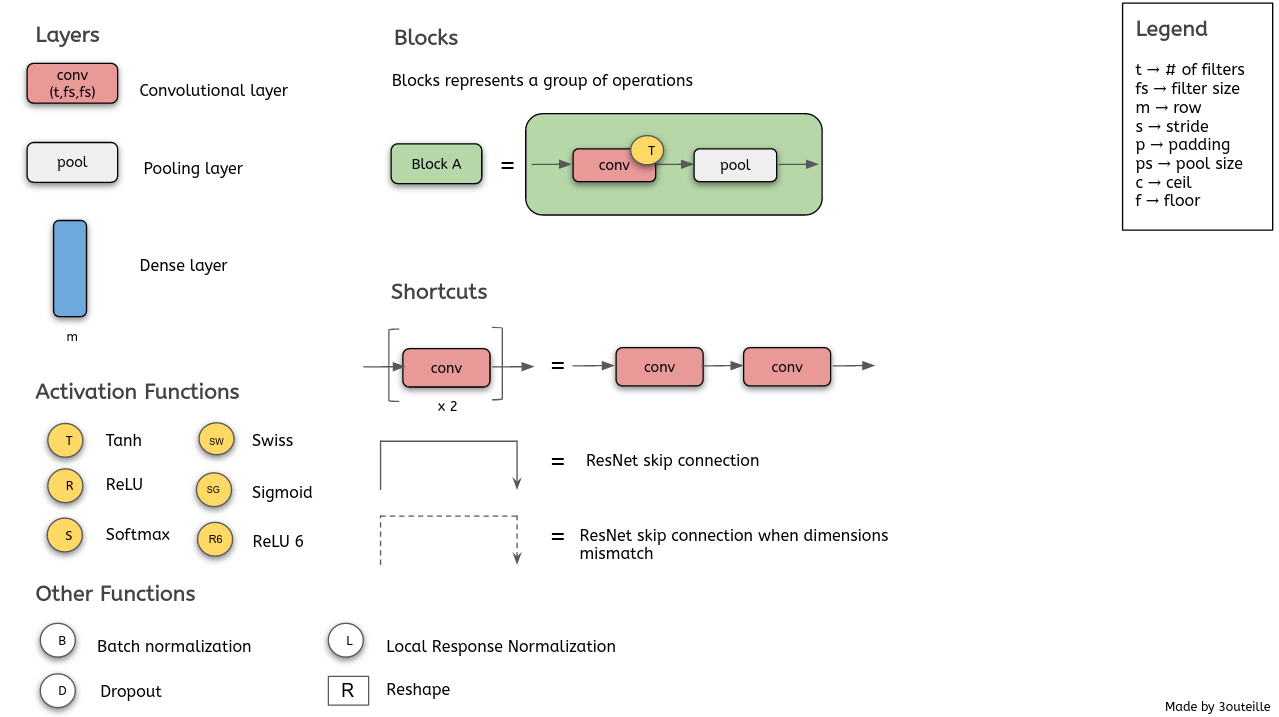
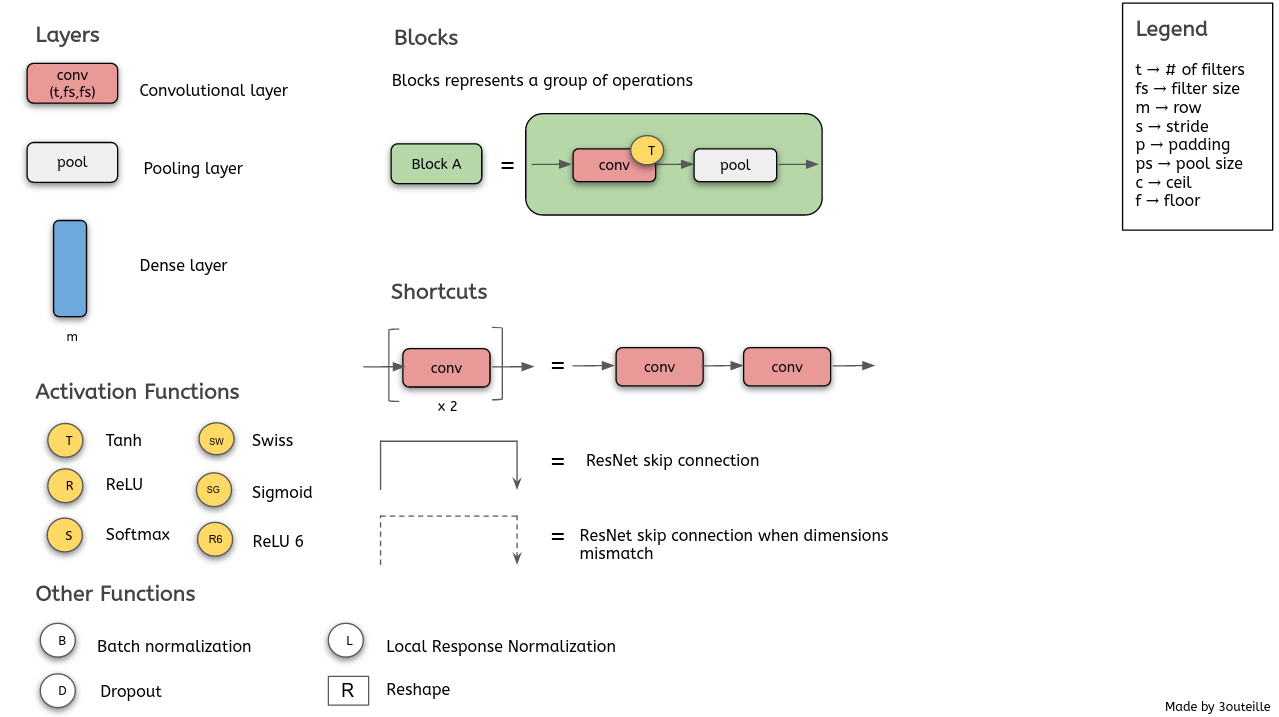
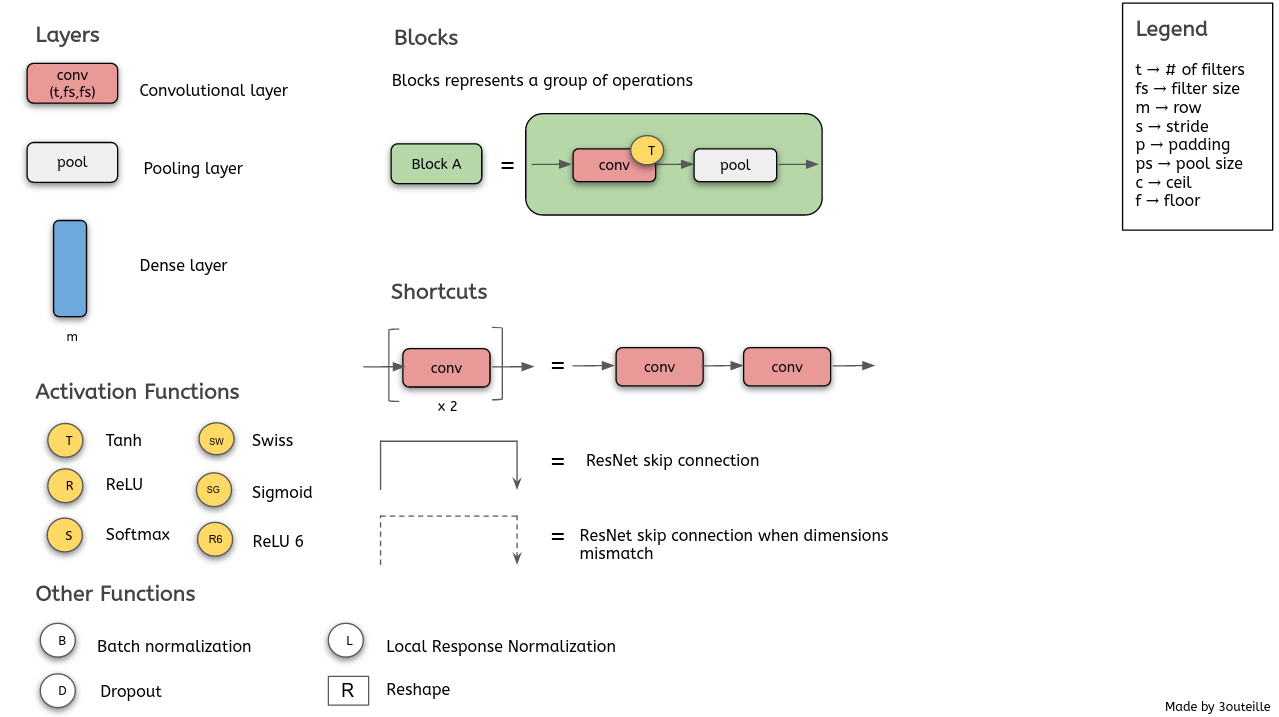
1. **Augmentation**

-----------------------------------------------------------------------------------------------------------------



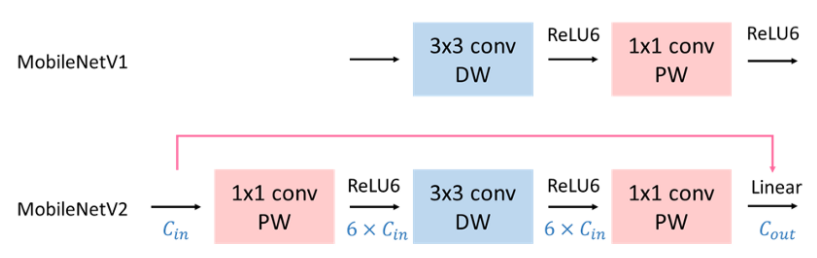
-----------------------------------------------------------------------------------------------------------------

1. **Model**

* Load in the pretrained base model, **MobileNetV2** network
* Unfreeze the base model and set the bottom layers to be un-trainable
* **MobileNetV2**

Reference: <https://hackmd.io/@bouteille/ryaDuxe5L>

* MobileNetV2 architecture is based on an **inverted residual structure** where the shortcut connections are between the **thin bottleneck layers**.
* It results in a very memory-efficient inference model.
* Comparison between MobileNetV1 and MobileNetV2:



* The bottleneck residual block is very important, which improves the state of the art performance of mobile models on multiple tasks and benchmarks as well as across a spectrum of different model sizes.

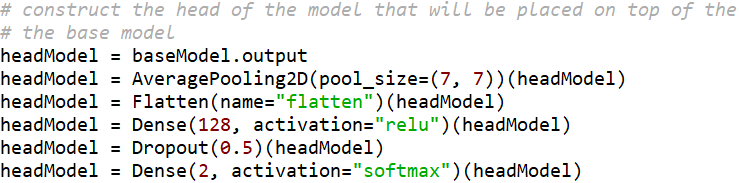
-----------------------------------------------------------------------------------------------------------------

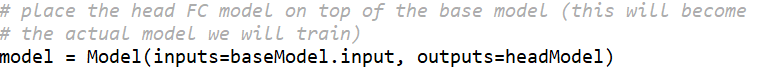


-----------------------------------------------------------------------------------------------------------------

* Stack the classification layers on top (Add a classification head)
* Softmax output **2 classes**: with mask and without mask

-----------------------------------------------------------------------------------------------------------------



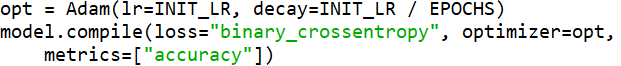


-----------------------------------------------------------------------------------------------------------------

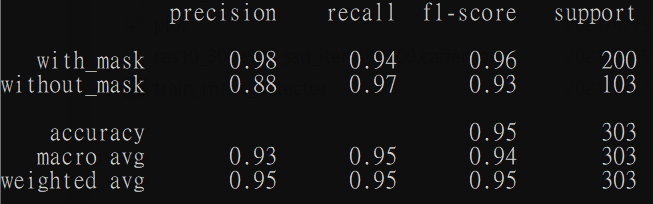
1. **Train and testing**

* Compile model, learning rate=1e-4, epoch=20, batch size=32

-----------------------------------------------------------------------------------------------------------------



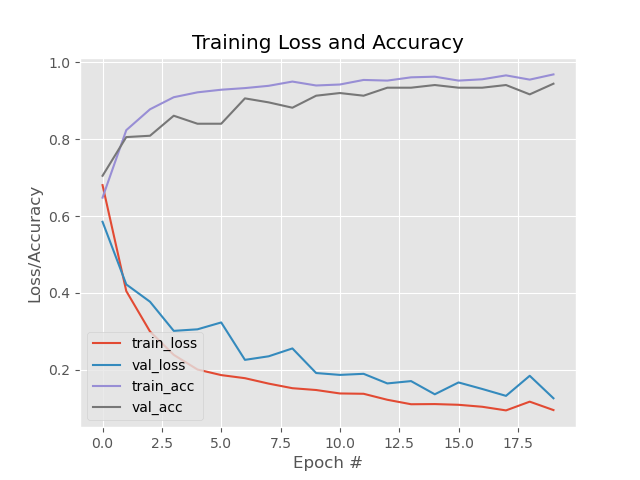
-----------------------------------------------------------------------------------------------------------------

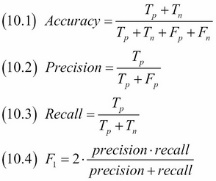


Training = 1211 images

Testing = 303 images

(200 mask, 103 no mask)





from sklearn.metrics import classification\_report

<https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html>

**STEP2. Detect mask image**

* Use two model:

1. **Caffe model**, a deep learning framework for object detectionto **detect face**.
2. **Mask detector model**, we use transfer learning with **MobileNetV2** as base model, to **detect wearing mask or not**.

* Step:

1. Load two model
2. Load the input image, resize it
3. Use Caffe model to detect face, draw bounding box
4. Use Mask detector model to determine if the face has a mask or not