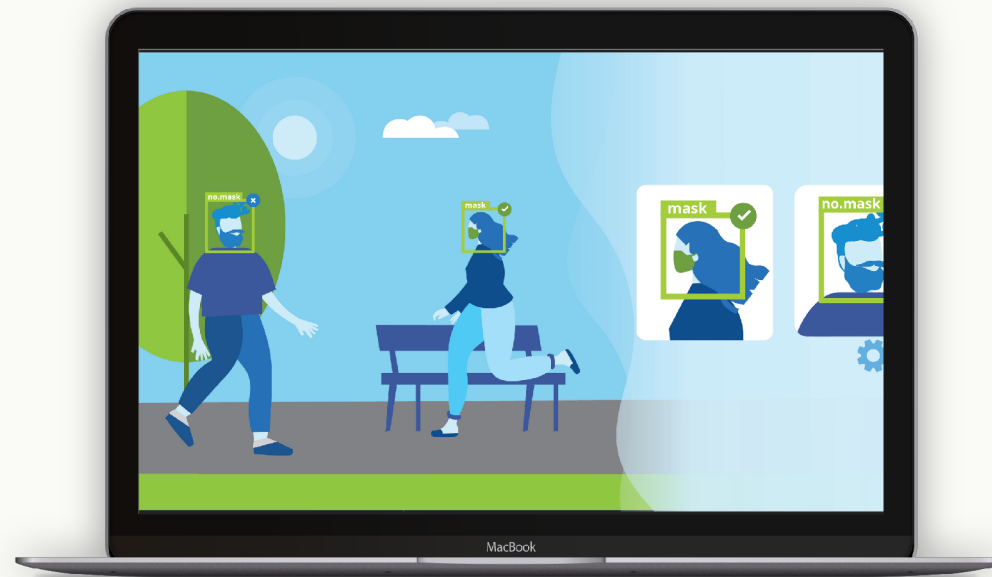


# Face Mask Detection



Contributors : Sohan Angelo - 180929158, Shravan Raviraj - 180929196

# Task : Binary Classification of Images

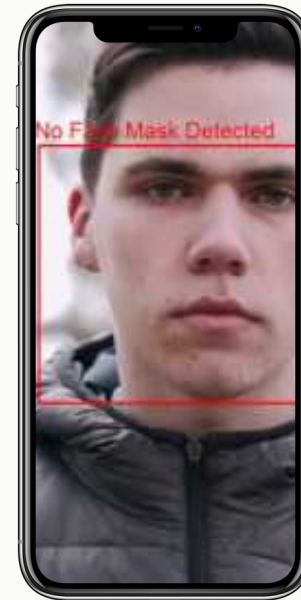
**Problem Statement :** Classify Images belonging to two categories -> "With Mask" & "Without Mask"

**Current State-of-the-Art :** Convolutional Neural Network Architectures

**Networks of choice :** Inception-v3 Network, MobileNet-v2 Network



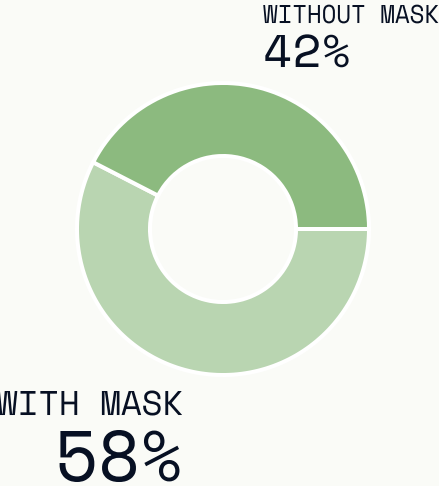
With Mask



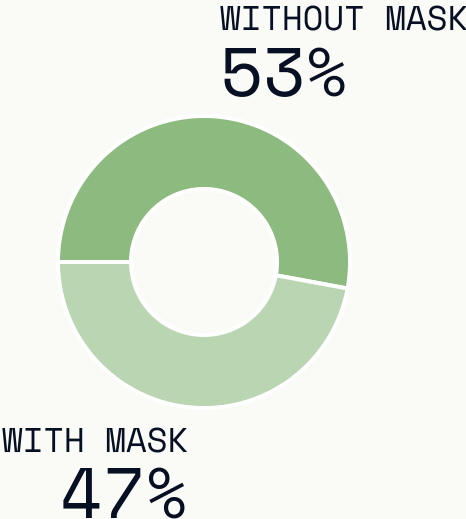
Without Mask

# Dataset Description

## TRAINING DATA



## TESTING DATA



LABEL	TRAIN	TEST
WITH MASK	594	154
WITHOUT MASK	438	173
TOTAL	1,032	397

# EDGE IMPULSE - Data Acquisition


The screenshot displays the Edge Impulse Data Acquisition web interface. The left sidebar contains navigation links: Dashboard, Devices, Data acquisition, Impulse design, Create impulse, Image, Transfer learning, Retrain model, Live classification, Model testing, Versioning, and Deployment. The main content area is titled 'DATA ACQUISITION (IoT)' and features tabs for 'Training data' and 'Test data'. A notification bar states: 'Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options'. Below this, two summary cards show 'DATA COLLECTED 1,032 items' and 'LABELS 2'. A table titled 'Collected data' lists 11 samples with columns for Sample Name, Label, Added, and Length. All labels are 'without mask' and added 'Yesterday, 17:55:28'. A 'Record new data' section on the right includes a 'Connect using WebUSB' button and a message: 'No devices connected to the remote management API.' Below this is a 'RAW DATA' section with a button to 'Click on a sample to load...'.

SAMPLE NAME	LABEL	ADDED	LENGTH
6.jpg.27ldvv12	without mask	Yesterday, 17:55:28	-
7.jpg.27ldvv0v	without mask	Yesterday, 17:55:28	-
13.jpg.27ldvuvv	without mask	Yesterday, 17:55:28	-
1.jpg.27ldvuse	without mask	Yesterday, 17:55:28	-
0.jpg.27ldvuts	without mask	Yesterday, 17:55:28	-
3.jpg.27ldvus5	without mask	Yesterday, 17:55:28	-
2.jpg.27ldvuse	without mask	Yesterday, 17:55:28	-
4.jpg.27ldvuro	without mask	Yesterday, 17:55:28	-
9.jpg.27ldvuqp	without mask	Yesterday, 17:55:28	-
10.jpg.27ldvunj	without mask	Yesterday, 17:55:28	-
11.jpg.27ldvun7	without mask	Yesterday, 17:55:28	-

**Dataset was prepared by collecting images from various online sources and was uploaded onto Edge Impulse**

# IMPULSE DESIGN

**Image data**




**Axes**  
image

Image width  
96

Image height  
96


Resize mode  
Fit shortest axis



**i** For optimal accuracy with transfer learning blocks, use a 96x96 or 160x160 image size.



**Image**




Name  
Image

Input axes  
☒ image



**Transfer Learning (Images)**



Name  
Transfer learning

Input features  
☒ Image

Output features  
2 (with mask, without mask)

Output features

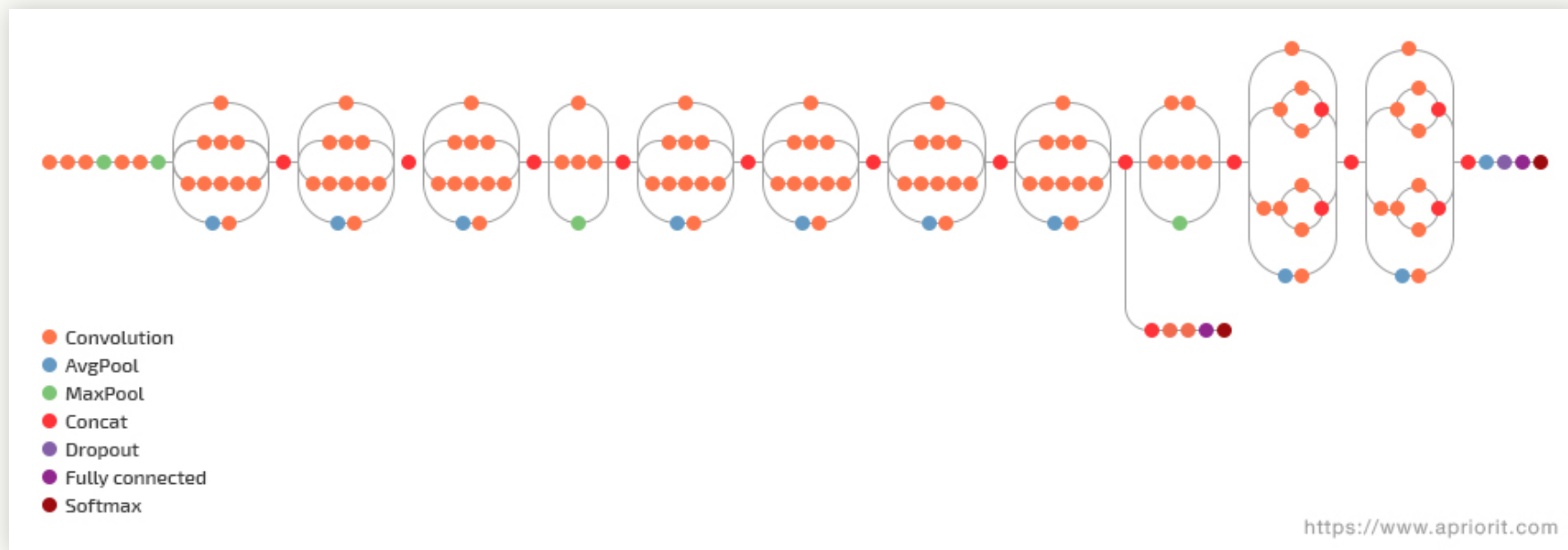


2 (with mask, without mask)

# Transfer Learning - Architecture

## Inception-v3

Inception is created to serve the purpose of reducing the computational burden of deep neural nets while obtaining state-of-art performance. As the network goes deeper, the computational efficiency will also decrease, therefore the authors of Inception were interested in finding a solution to scale up neural nets without increasing computational cost.



# Training Results

## Inception-v3

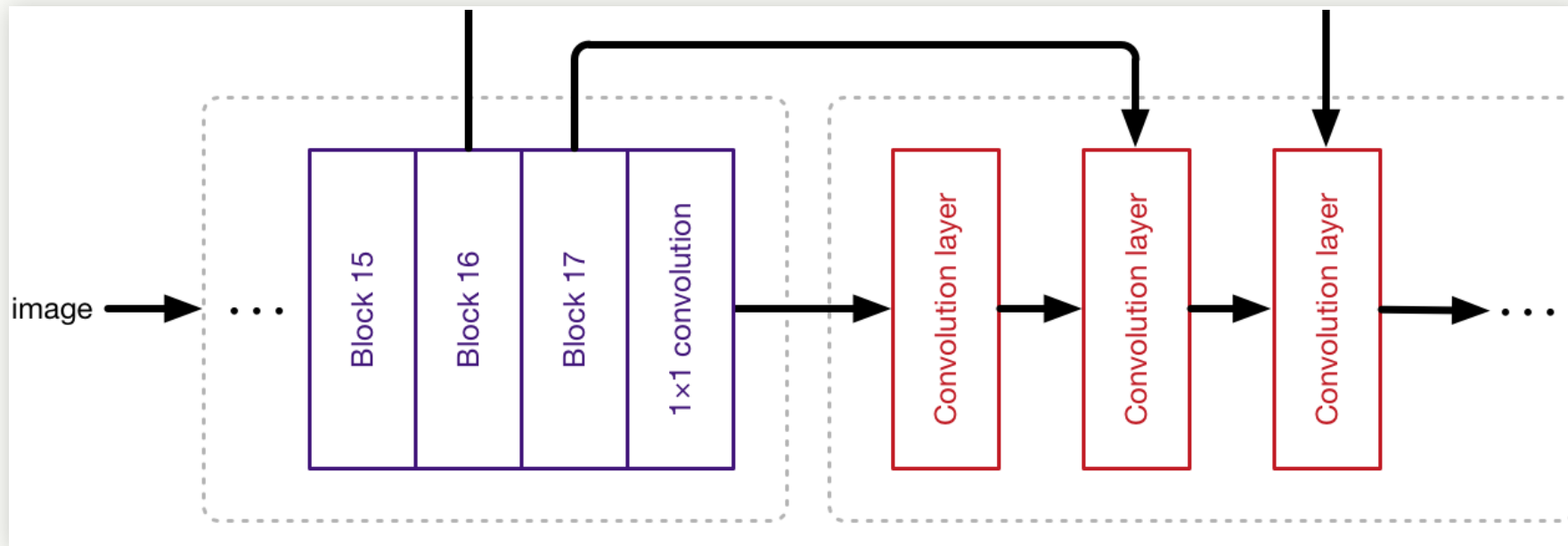
```
Epoch 1/6
51/51 [=====] - 107s 2s/step - loss: 0.7256 - accuracy: 0.8227
Epoch 2/6
51/51 [=====] - 103s 2s/step - loss: 0.2156 - accuracy: 0.9241
Epoch 3/6
51/51 [=====] - 98s 2s/step - loss: 0.1802 - accuracy: 0.9429
Epoch 4/6
51/51 [=====] - 98s 2s/step - loss: 0.1781 - accuracy: 0.9527
Epoch 5/6
51/51 [=====] - 98s 2s/step - loss: 0.1152 - accuracy: 0.9576
Epoch 6/6
51/51 [=====] - 106s 2s/step - loss: 0.1188 - accuracy: 0.9616
```

**Accuracy =  
96.16%**

# Transfer Learning - Architecture

## MobileNet-v2

MobileNetV2 is a convolutional neural network architecture that seeks to perform well on mobile devices. It is based on an inverted residual structure where the residual connections are between the bottleneck layers. The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity. As a whole, the architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers.





# Training Results

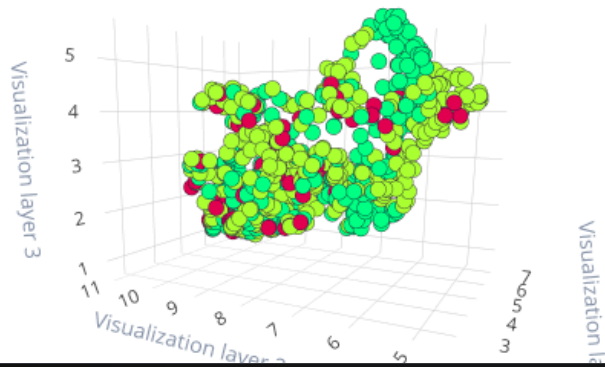
## MobileNet-v2

Confusion matrix (validation set)

	WITH MASK	WITHOUT MASK
WITH MASK	91.5%	8.5%
WITHOUT MASK	0%	100%
F1 SCORE	0.96	0.95

Feature explorer (full training set) ?

- with mask - correct
- without mask - correct
- with mask - incorrect



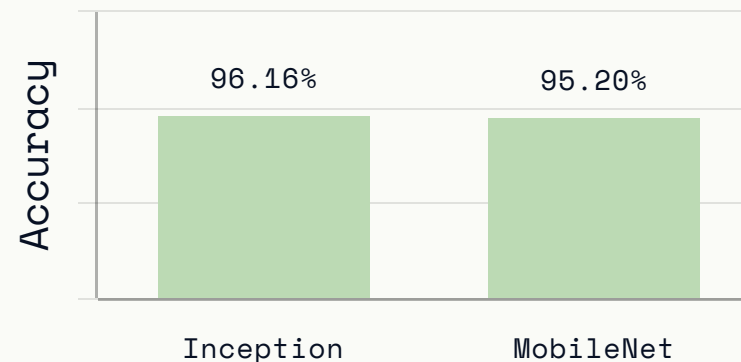
**Accuracy =  
95.2%**

# Transfer Learning - Architecture

## MobileNet-v2 vs Inception-v3

The major difference between InceptionV3 and Mobilenet is that Mobilenet uses Depthwise separable convolution while Inception V3 uses standard convolution. This results into lesser number of parameters in MobileNet compared to InceptionV3. However, this results in slight decrease in the performance as well.

The trend in the computer vision is to make deeper and more complicated network to achieve higher accuracy. However, deeper networks come with the tradeoff of size and speed. In real applications such as an autonomous vehicle or robotic visions, the object detection task must be able to be done on the computationally limited platform. MobileNet is developed to solve this problem, which is a network for embedded vision applications and mobile devices.



## MobileNet-v2 was Implemented with the Keras Expert Mode on the Edge Impulse Platform

### Neural network architecture

```
1
2 import tensorflow as tf
3 from tensorflow.keras import Model
4 from tensorflow.keras.models import Sequential
5 from tensorflow.keras.layers import Dense, InputLayer, Dropout, Conv1D, Flatten, Reshape, MaxPooling1D
6 from tensorflow.keras.optimizers import Adam, Adadelta
7 from tensorflow.keras.losses import categorical_crossentropy
8
9 # Load best model from initial training
10 sys.path.append('./resources/libraries')
11 import ei_tensorflow.training
12
13 INPUT_SHAPE = (96, 96, 3)
14
15 base_model = tf.keras.applications.MobileNetV2(
16     input_shape=INPUT_SHAPE, alpha=0.35,
17     weights='./transfer-learning-weights/keras/mobilenet_v2_weights_tf_dim_ordering_tf_kernels_0.35_96
18     .h5',
19     include_top=True
20 )
21
22 base_model.trainable = False
23
24 model = Sequential()
25 model.add(InputLayer(input_shape=INPUT_SHAPE, name='x_input'))
26 # Don't include the base model's top layers
27 last_layer_index = -3
28 model.add(Model(inputs=base_model.inputs, outputs=base_model.layers[last_layer_index].output))
29 model.add(Dense(16))
30 model.add(Dropout(0.1))
31 model.add(Flatten())
32 model.add(Dense(classes, activation='softmax'))
```

# DEMONSTRATION

## PLATFORM: EDGE IMPULSE

# NEURAL NETWORK ARCHITECTURE : MobileNet-v2

## DATA ACQUISITION DEVICE : Mobile Phone

## Sensor : Camera

