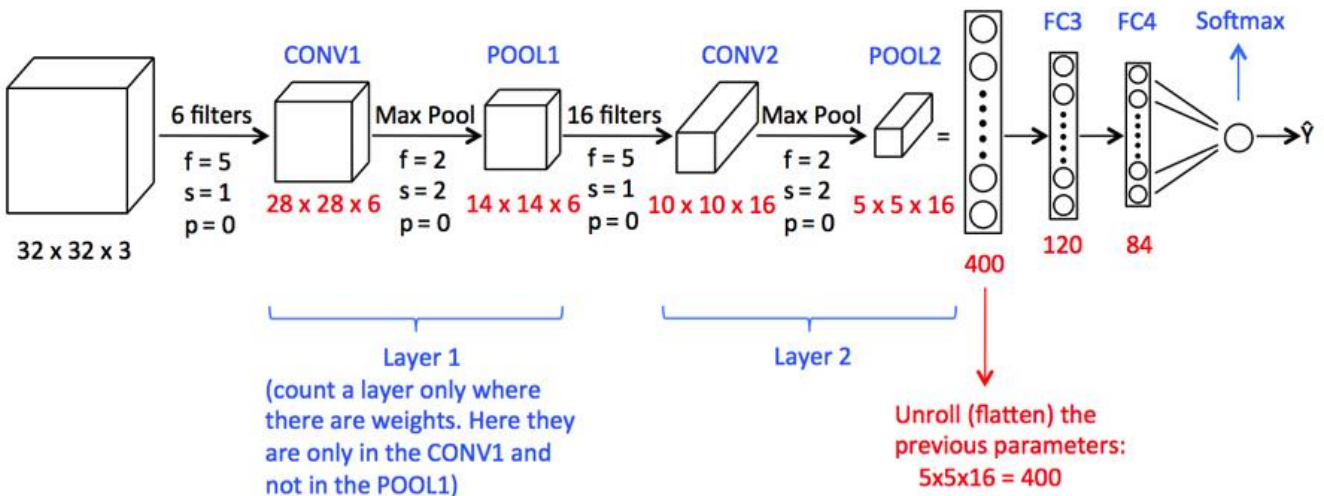


## Selected-2 Project Documentation

### 1) Architecture used in the paper:



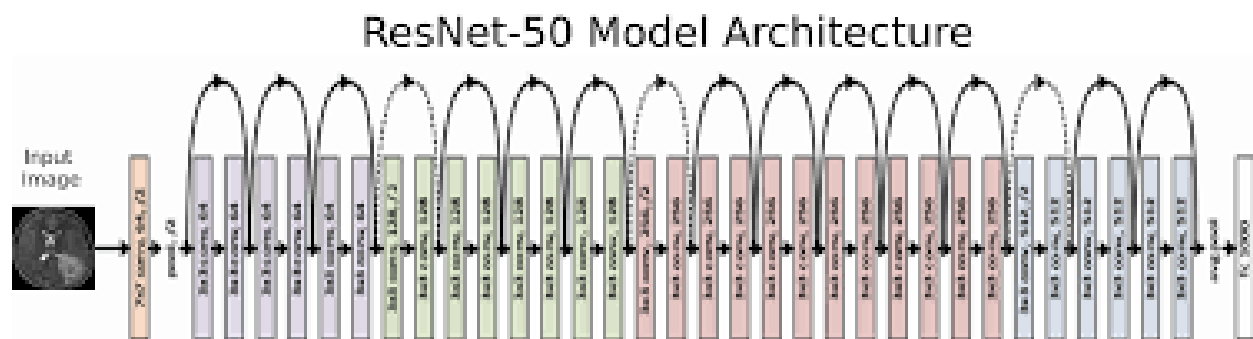
-The first thing you should do is feed the pixels of the image in the form of arrays to the input layer of the neural network (MLP networks used to classify such things). The hidden layers carry Feature Extraction by performing various calculations and operations. There are multiple hidden layers like the convolution, the ReLU, and the pooling layer that performs feature extraction from your image. So finally, there is a fully connected layer that you can see which identifies the exact object in the image.

## 2) Dataset Details:

- The dataset detects different car brand based on image classification.

## 3) Implementation Details:

- we are going to import the necessary libraries which are required for performing CNN tasks.
- we are using res.net architecture



ResNet50 is a variant of ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has  $3.8 \times 10^9$  Floating points operations. It is a widely used ResNet model and we have explored ResNet50 architecture in depth.

### -We make a model summary

Total params: 23,888,771

Trainable params: 301,059

Non-trainable params: 23,587,712

-We are required to specify optimizers.

```
13]: # tell the model what cost and optimization method to use
model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)
```

-We will see how to set the data directory and generate image data.

```
] : # Make sure you provide the same target size as initialied for the image size
training_set = train_datagen.flow_from_directory('D:/computer science/level3/semester 2/selected 2/selected2/Datasets/Train',
                                                target_size = (224, 224),
                                                batch_size = 32,
                                                class_mode = 'categorical')
```

Found 64 images belonging to 3 classes.

```
] : test_set = test_datagen.flow_from_directory('D:/computer science/level3/semester 2/selected 2/selected2/Datasets/Test',
                                                target_size = (224, 224),
                                                batch_size = 32,
                                                class_mode = 'categorical')
```

Found 69 images belonging to 3 classes.

-Final step of the fitting model.

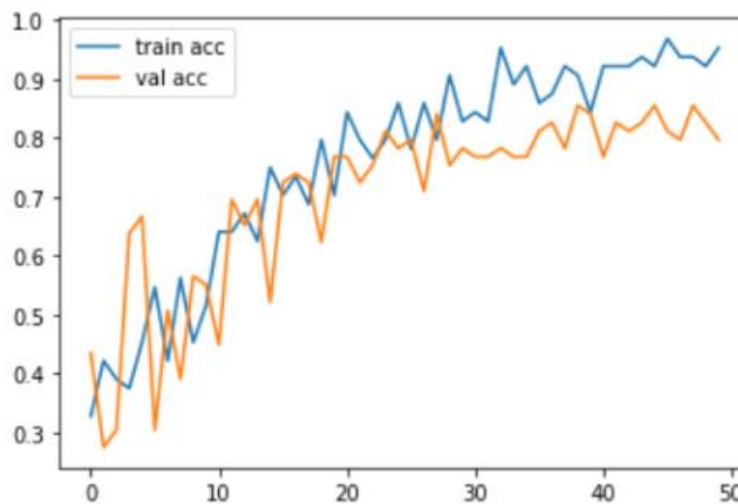
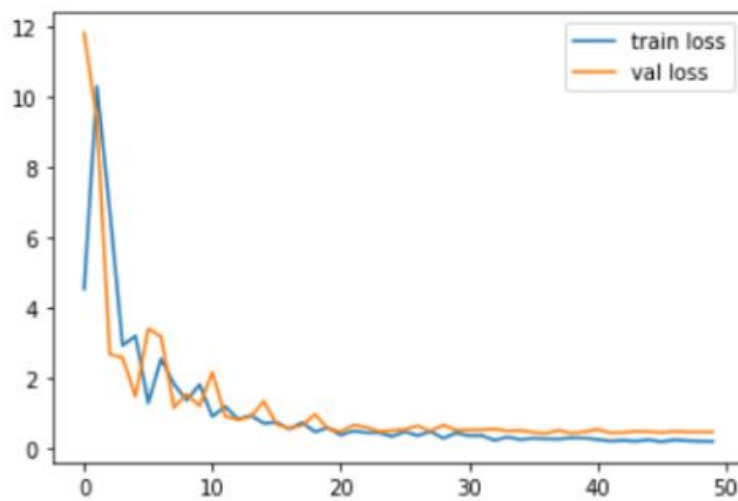
```
] : # fit the model
# Run the cell. It will take some time to execute
r = model.fit(
    training_set,
    validation_data=test_set,
    epochs=50,
    steps_per_epoch=len(training_set),
    validation_steps=len(test_set)
)
```

## 4) Results and Visualization:

### - Results:

Epoch 50/50  
2/2 [=====] - 6s 5s/step - loss: 0.2157 - accuracy: 0.9531 - val\_loss: 0.4833 - val\_accuracy: 0.7971

### - Visualization:



- **Feature Extracting:**

