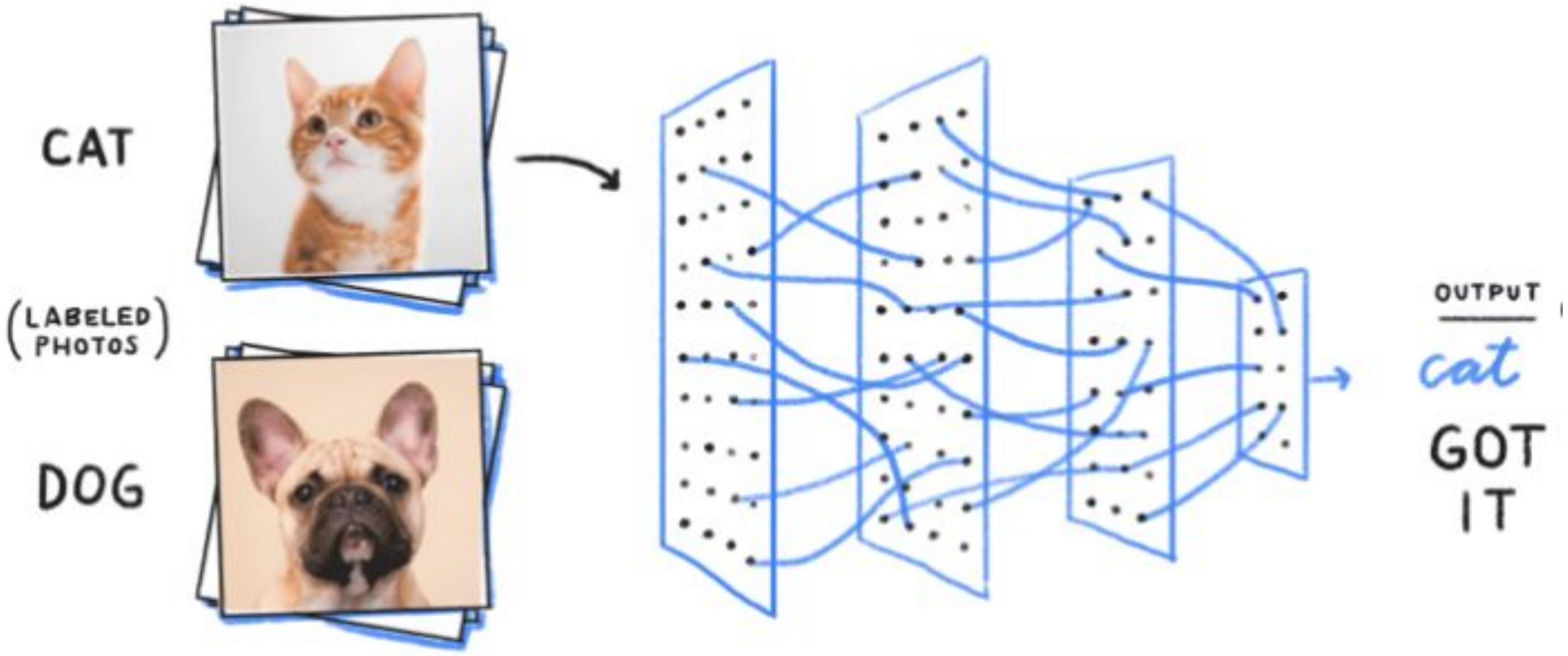
IMAGE CLASSIFICATION
with CIFAR-10 dataset

Foundations of Deep Learning
Academic Year 2022/23

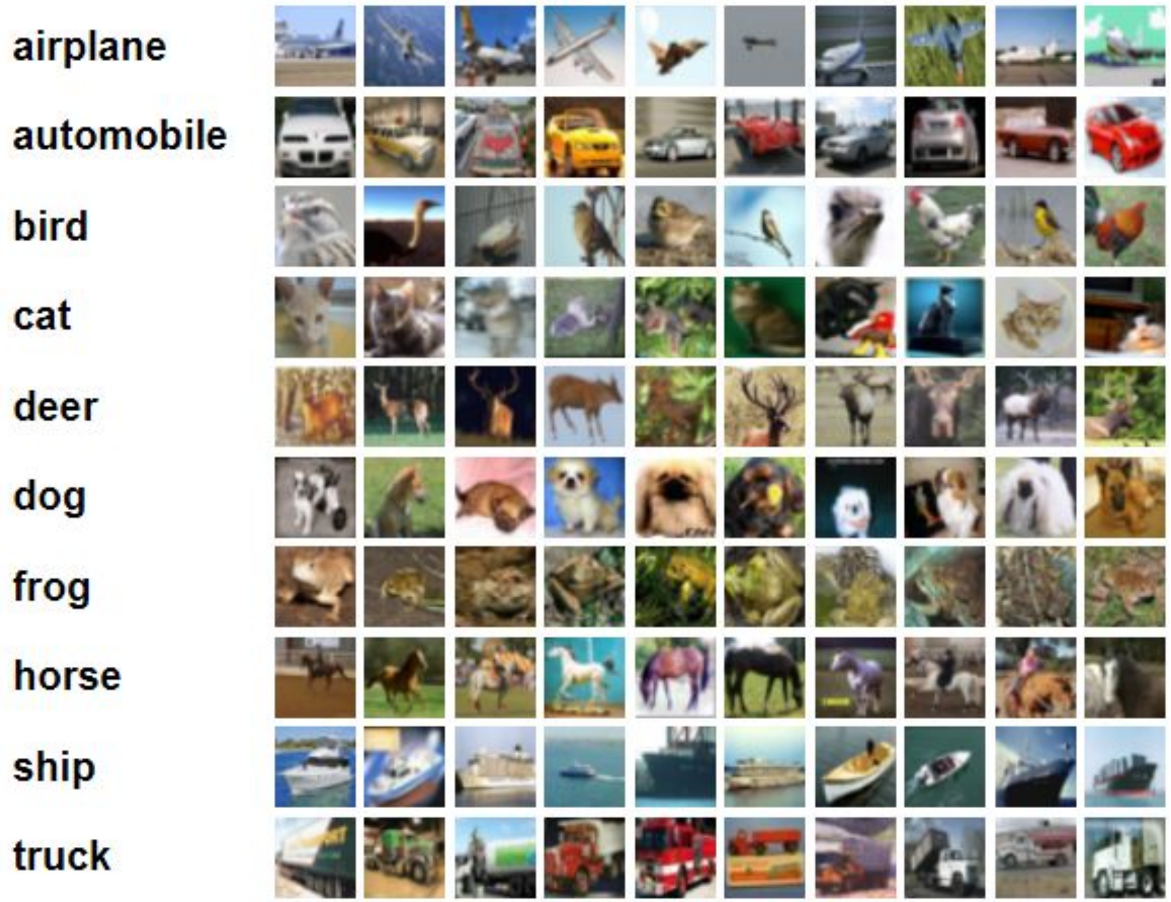
Salvatore Rastelli
Marco Sallustio
Marco Guarisco

INTRODUCTION & GOAL

The aim of the project is to develop a model that is able to classify different images in different classes.
In order to do so, we implemented a Convolutional Neural Network (CNN) in a way that will be explained in the following slides.



DATASET

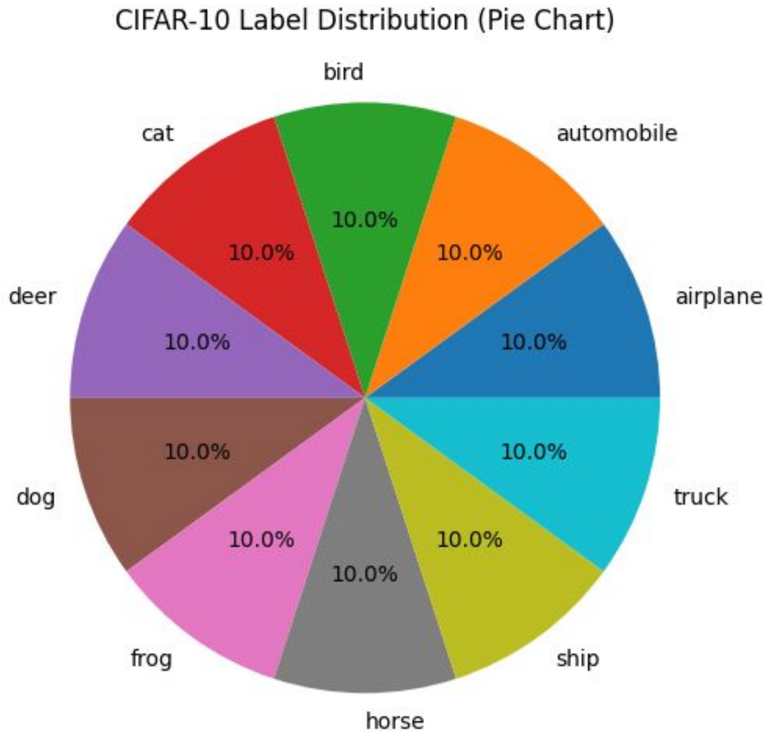


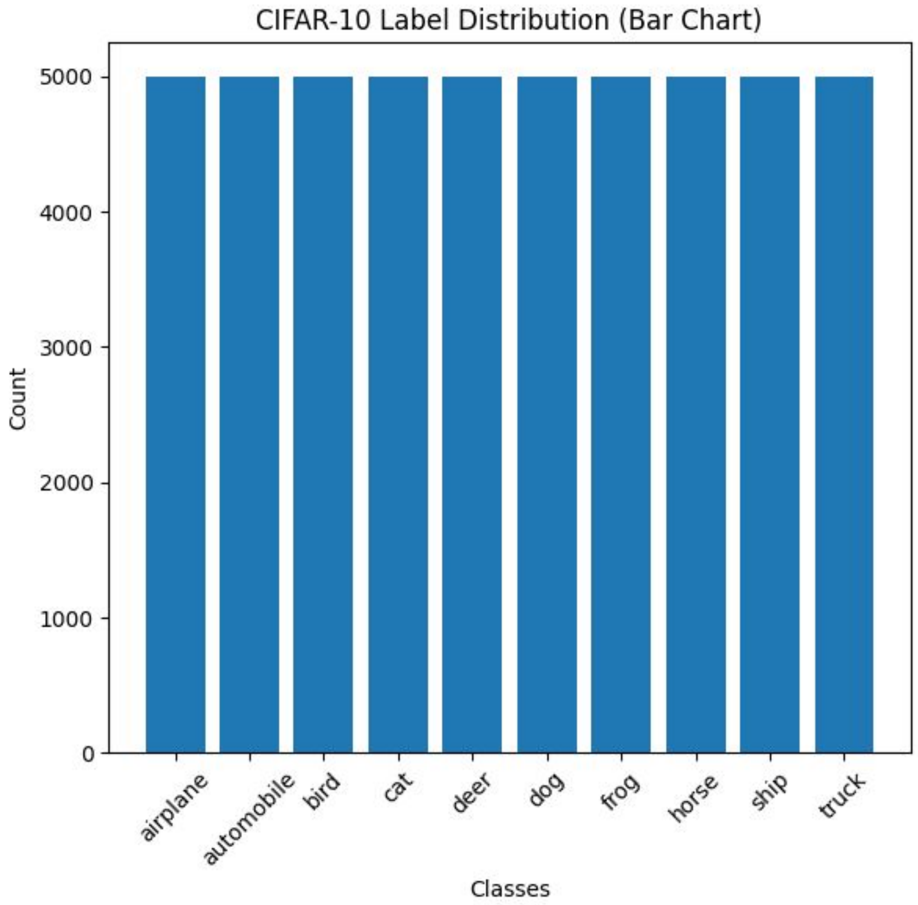
WHAT DOES IT INCLUDE?

The dataset consists of 60000 32x32 color images divided in 10 classes, which are mutually exclusive.
It is already divided in a **Training** set (composed of 50000 images) and a **Test** set (composed of 10000 images)

DESCRIPTIVE ANALYSIS

As we can see from both charts, all 10 classes are perfectly balanced, with exactly 5000 images per class in the Training set.

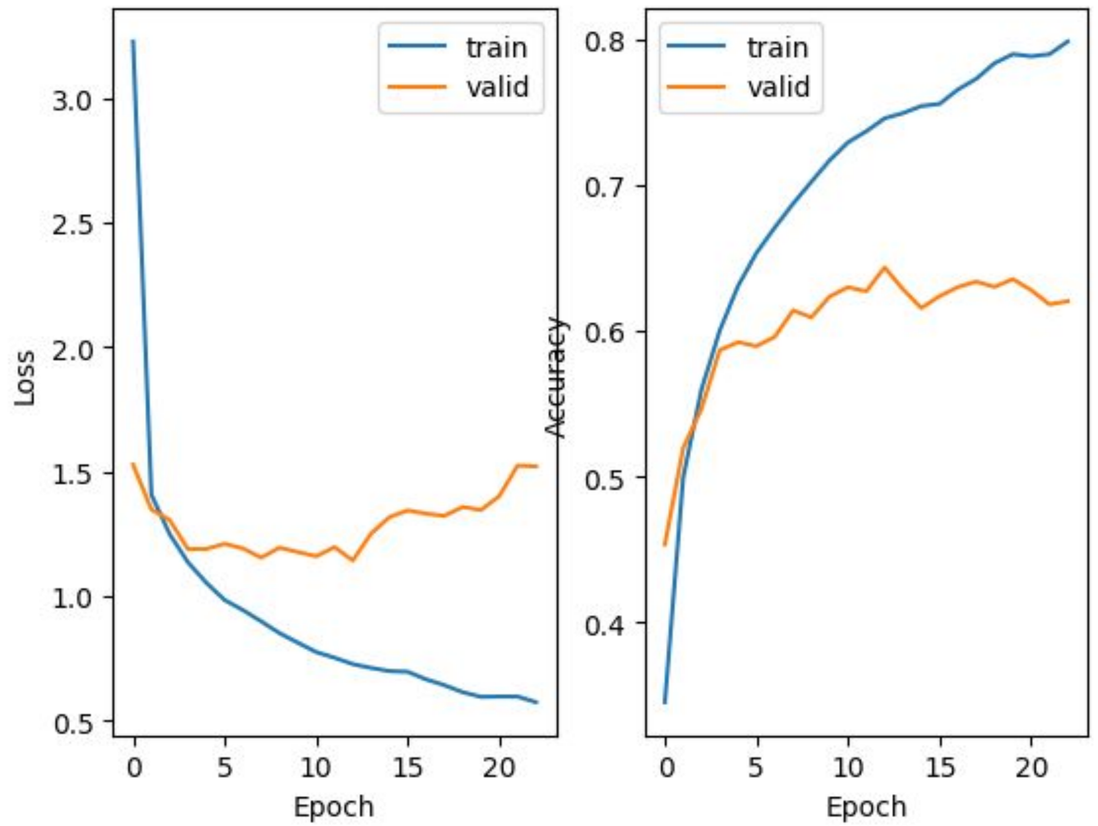




OUR SOLUTION: 1st ARCHITECTURE

HOW IS IT COMPOSED?

- **3 Convolutional layers** with *32*, *64* and *128* filters of dimension *3x3*
- **3 Activation layers** using *relu* function
- **2 MaxPooling2D layers** with window size *3x3* and stride of 3
- **1 GlobalMaxPooling2D layer**
- **1 Dense layer** with activation function *softmax*
- **Batch size** equal to 128
- *Adam* optimization with a *learning rate*=0,001
- **Loss function**: Categorical Cross Entropy



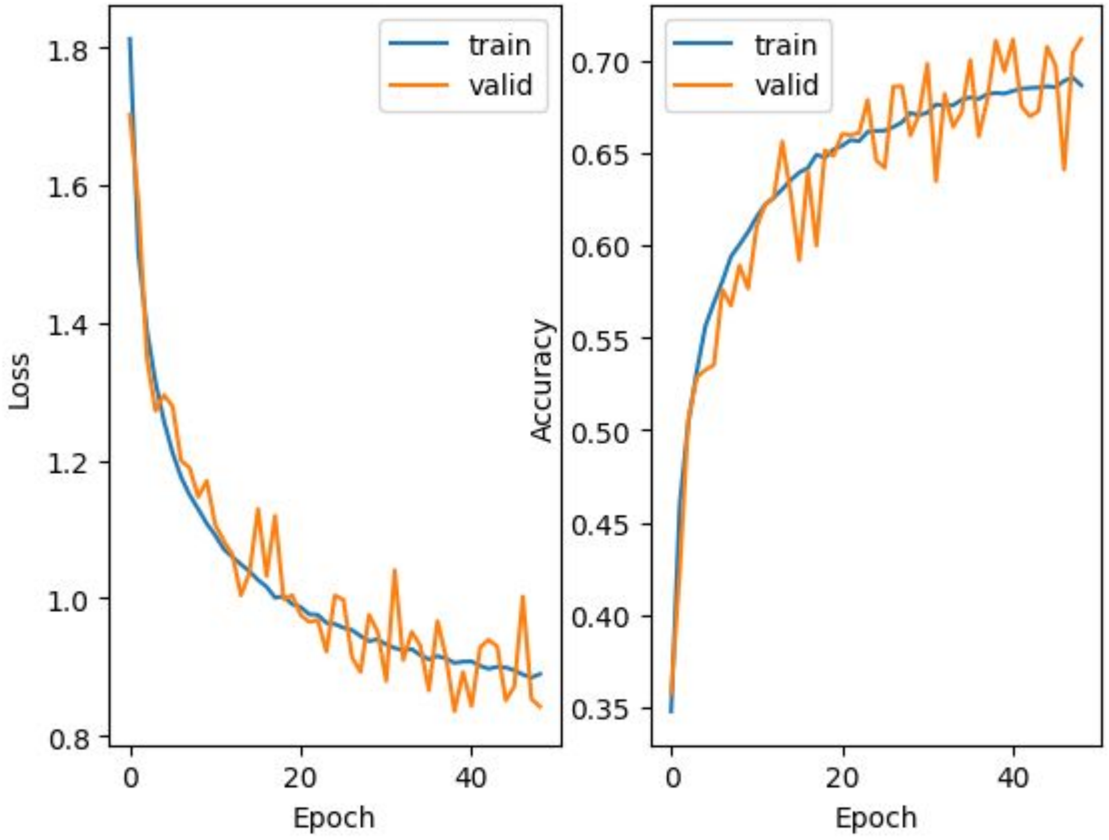
This model presents significant overfitting and low values of accuracy in the validation set.

OUR SOLUTION: 2nd ARCHITECTURE

HOW IS IT COMPOSED?

We improved the previous model by adding the following layers:

- **BatchNormalization**: applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1.
- **Dropout**: randomly sets input units to 0 with a preset frequency, which helps to prevent overfitting. It activate only during training.
- **Flatten**: remodels the tensor removing all the dimension but one.



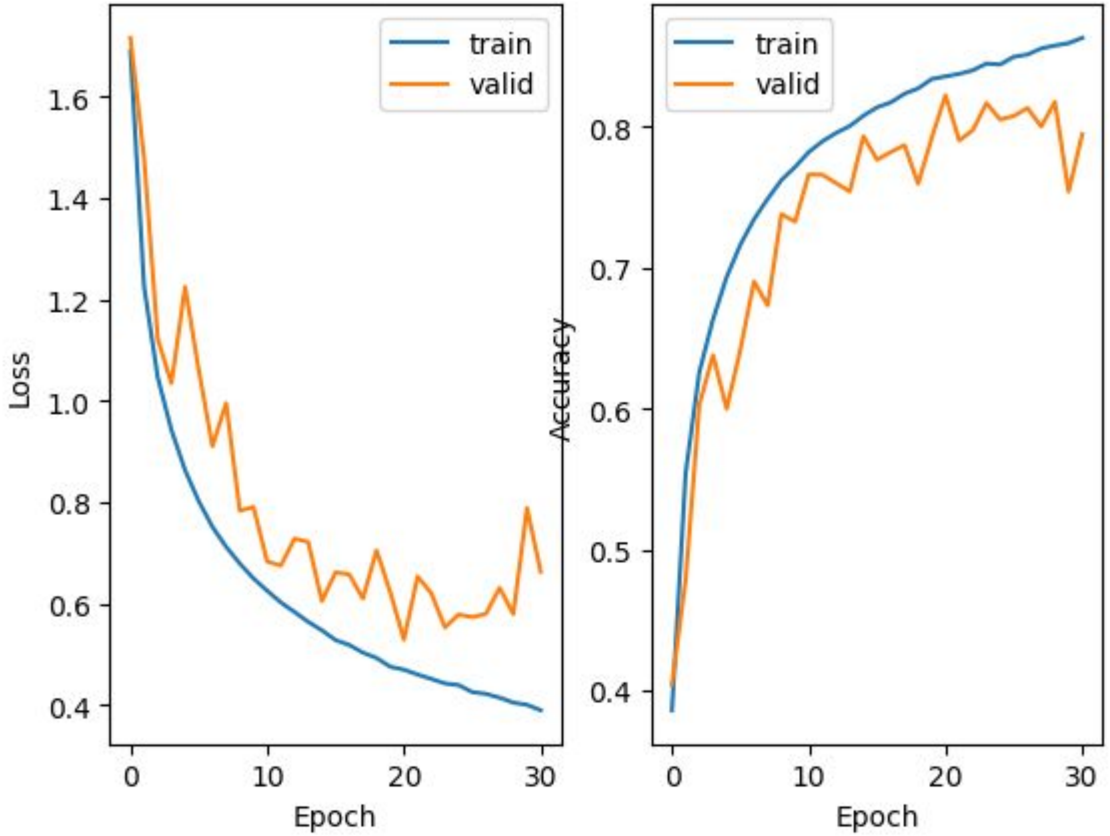
The introduction of these layers has reduced the overfitting, as well as improving the accuracy in the validation set. However, the accuracy is still low.

OUR SOLUTION: 3rd ARCHITECTURE

HOW IS IT COMPOSED?

In this architecture we have made the model more dense, by adding:

- **3 Convolutional layers** with *32* filters
- **3 Convolutional layers** with *64* filters.



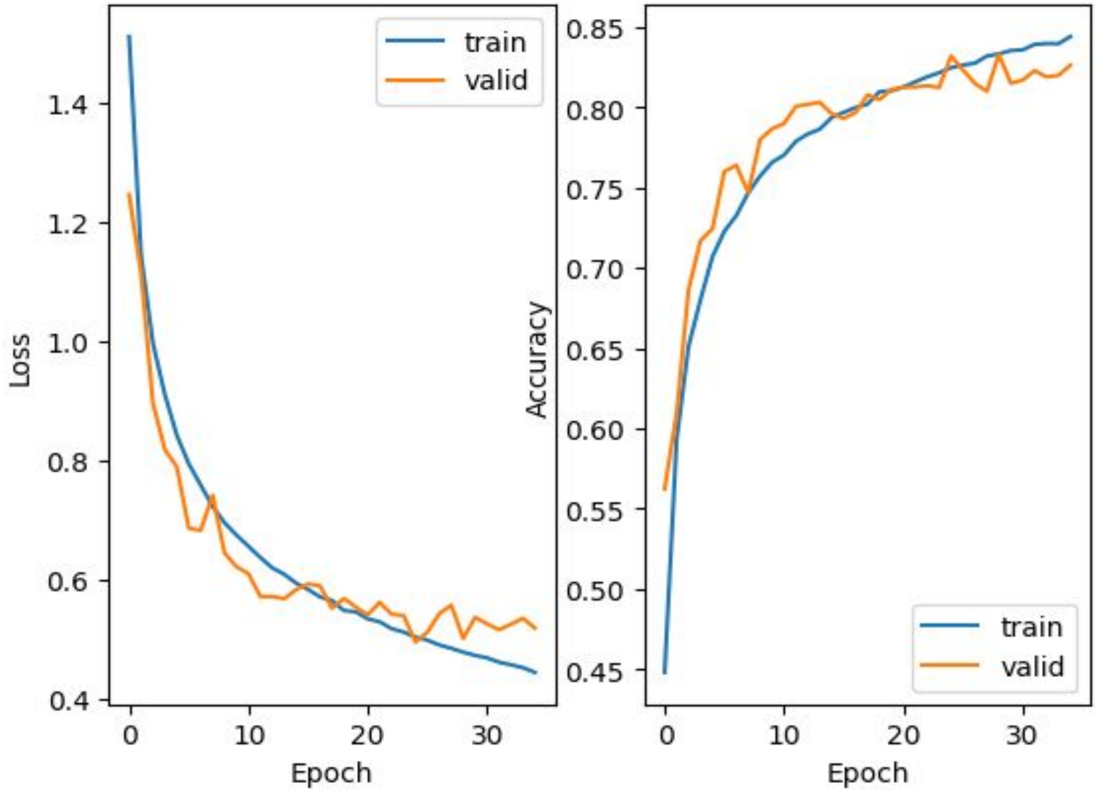
Making the model denser have improved the accuracy, but at the cost of an increased overfitting

OUR SOLUTION: 4^{th} ARCHITECTURE

HOW IS IT COMPOSED?

In this architecture we have modified the activation function in the 3^{rd} model, replacing the *relu* with the *Leaky ReLU*.
After the definition of the model, we now use the **GridSearchCV Algorithm** to identify the best Batch Size between the values 32, 64, 128, 256.
The algorithm suggested that the optimal batch size is **32**.

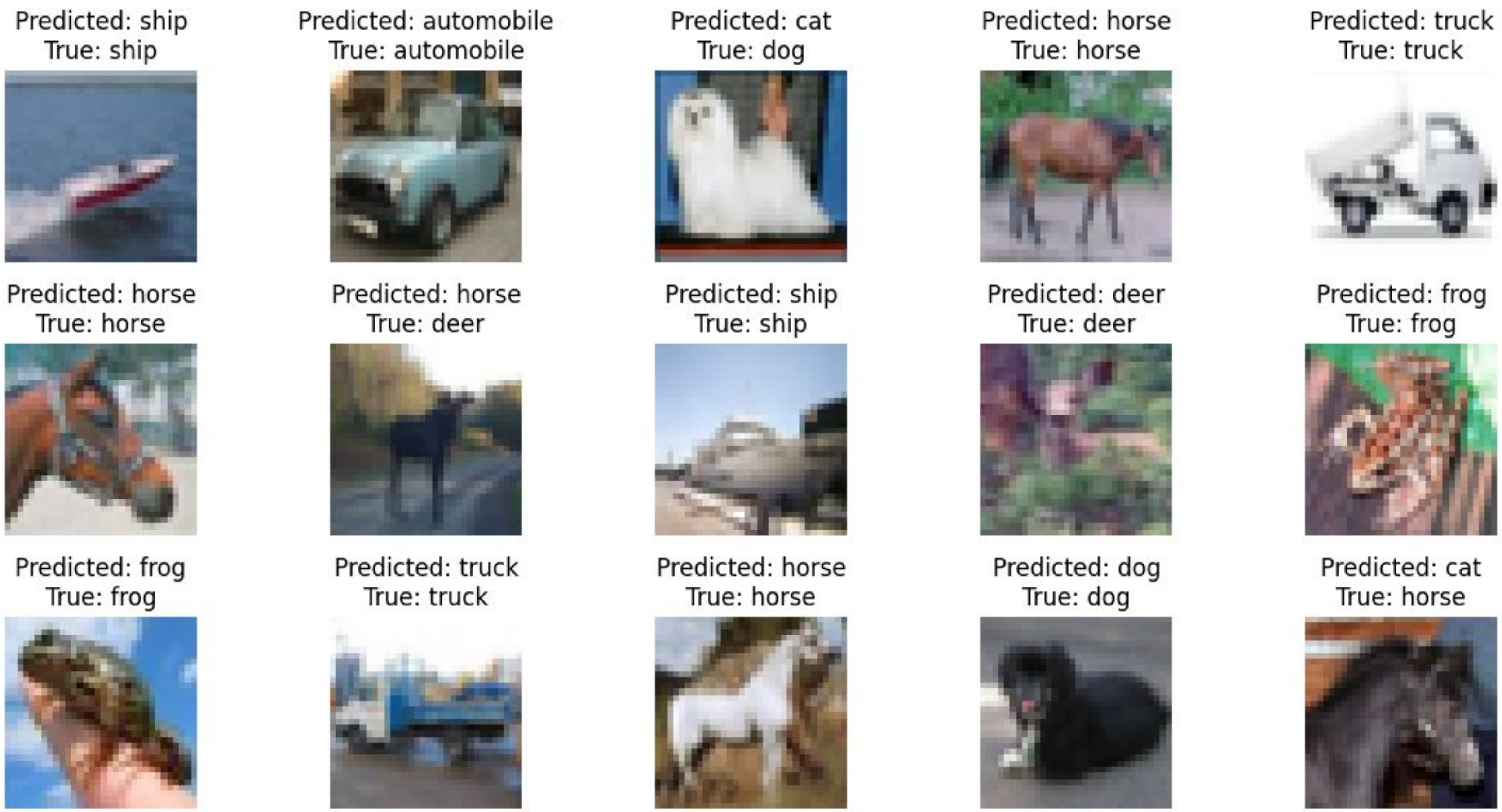
```
print(f'Best Accuracy using {grid_result.best_params_}')

Best Accuracy using {'batch_size': 32}
```



RESULTS

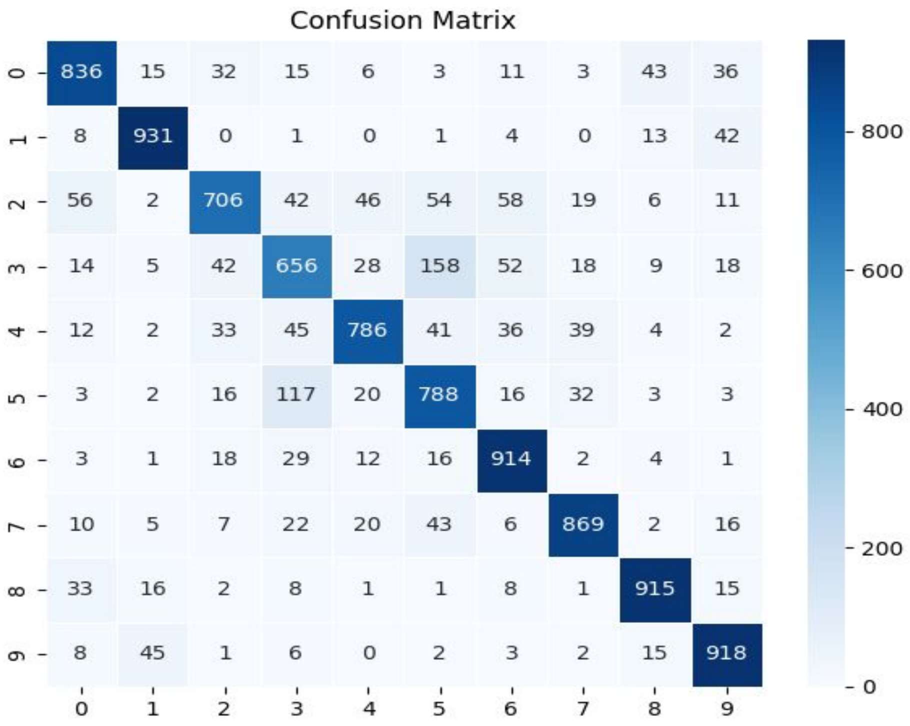
WHAT ARE THESE IMAGES?

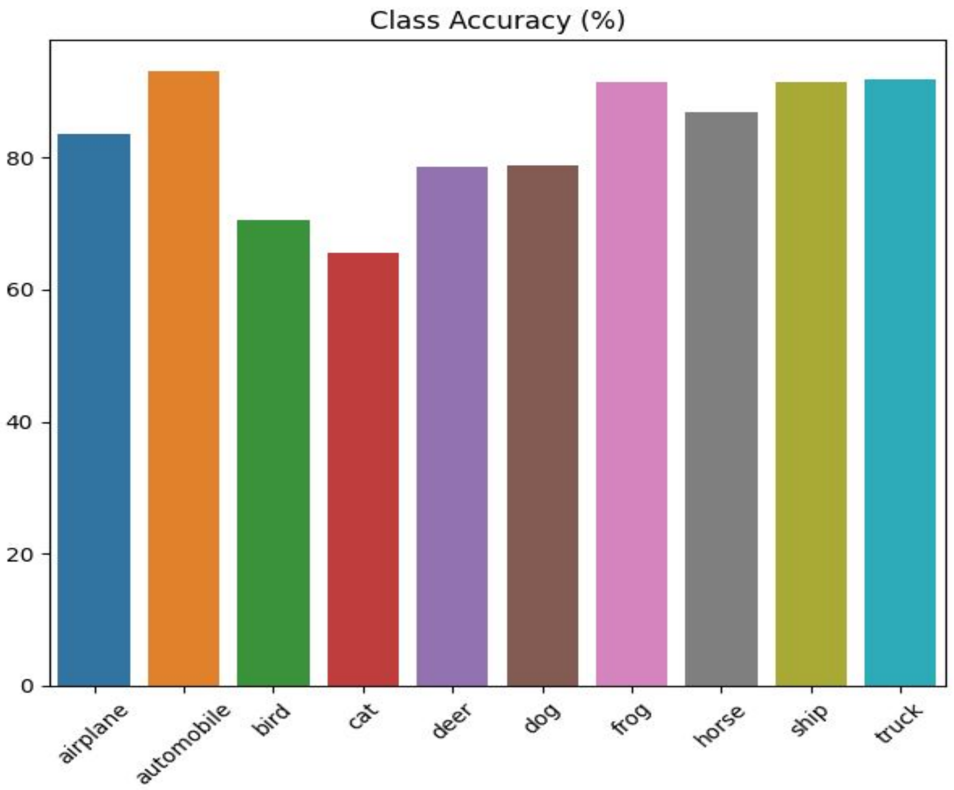
We evaluated our model on the test set and showed several images with the predicted label and the true label. In the following grid, 3 images out of 15 are incorrectly predicted.



PREDICTIONS

WHAT DO THEY TELL US?
The confusion matrix and the Accuracy percentage for each class are shown. The most accurately predicted class is *automobile*, while the least accurate is *cat*. From the plot we can see that the model struggles the most when trying to classify animals. In particular we can see that the highest number of occurrences of mislabeled images are 158 and 117, i.e. *dogs* mistaken for *cats* and vice-versa.



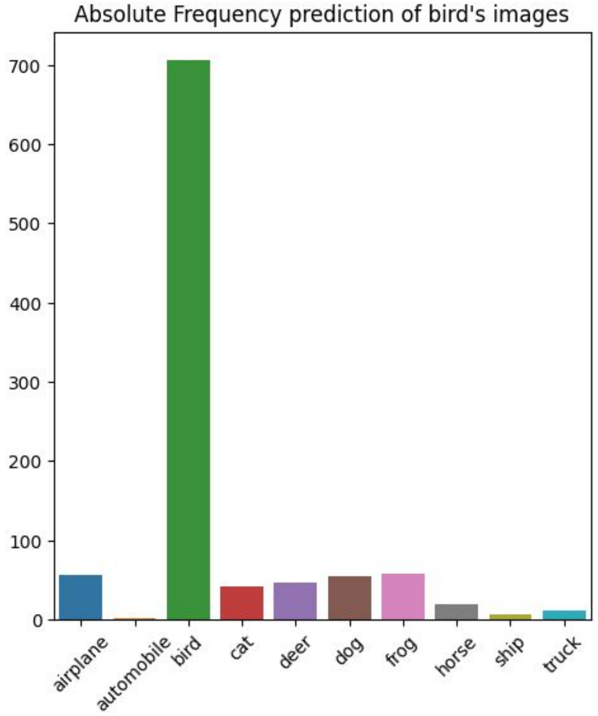


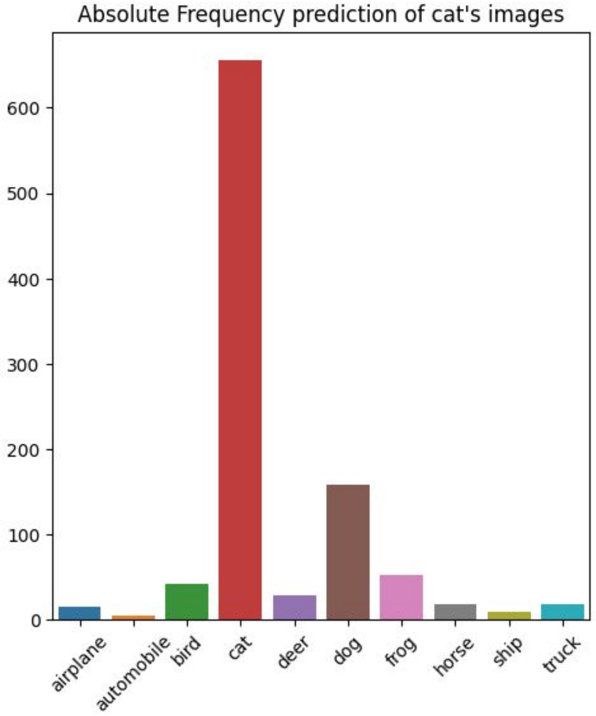
PREDICTIONS

WHAT DO THEY TELL US?

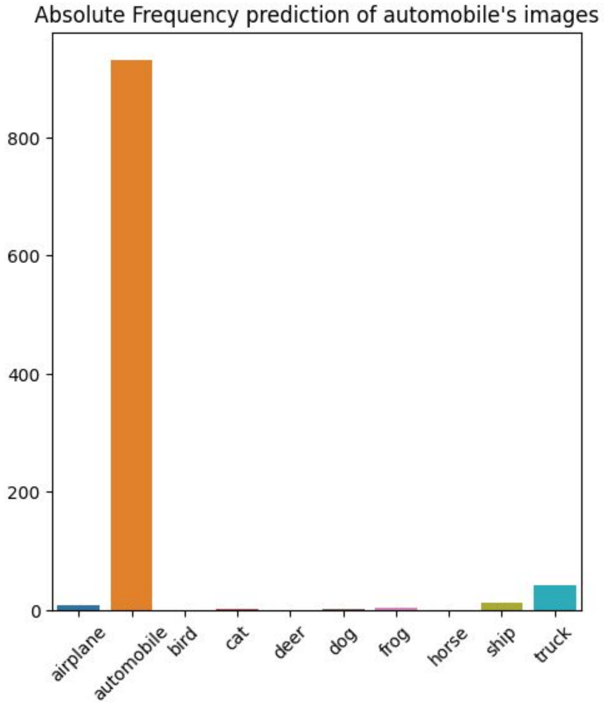
As mentioned earlier, the model struggles identifying **animals**, while it excels in identifying **vehicles**. Here we plot the predictions of the two worst (*birds* and *cats*) and the two best performing classes (*automobile* and *truck*).
On the animals side, we see that the model mainly mistakes *cats* for *dogs*, and wrongly labels bird pictures as another animal (*cat*, *deer*, *frog*), or as airplanes, approximately in equal proportions.
On the other hand, the model is very accurate in classifying *automobiles* and *trucks*. Nevertheless we see that sometimes it mislabels one for the other.

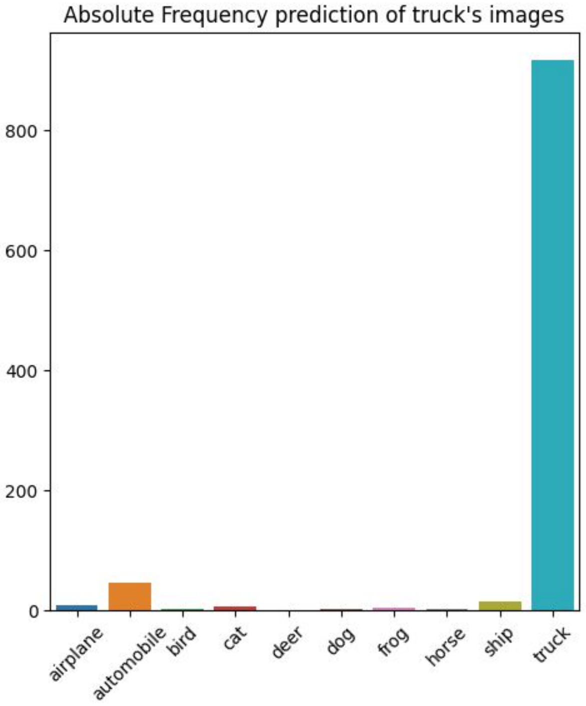
Animals





Vehicles

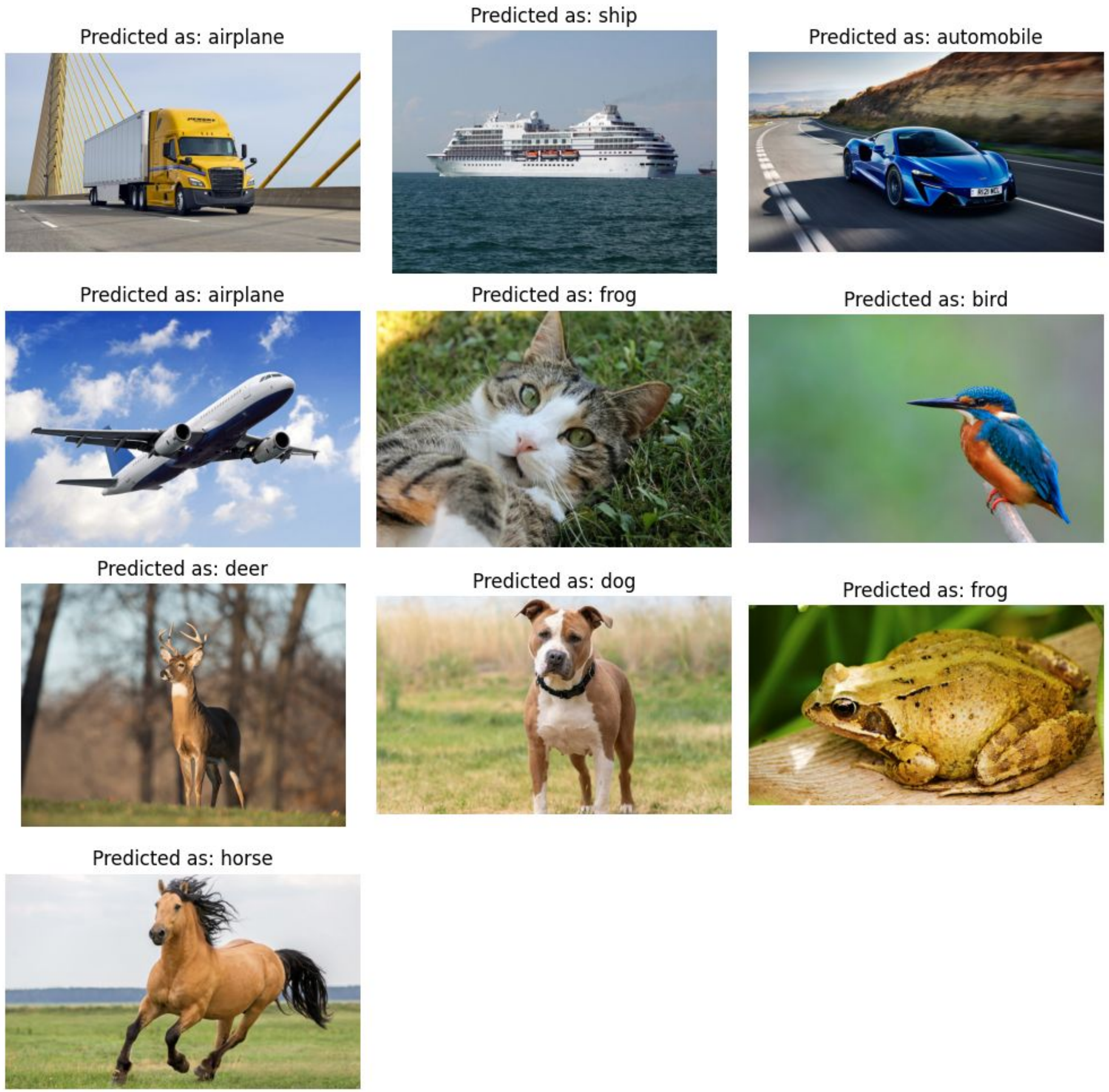




CLASSIFICATION ON NEW IMAGES

WHICH IMAGES HAVE BEEN UPLOADED?

To test the model, new images (one for each class) were taken from the web to find out if the neural network is able to classify them. 8 out of 10 images are correctly classified, which is the expected accuracy.

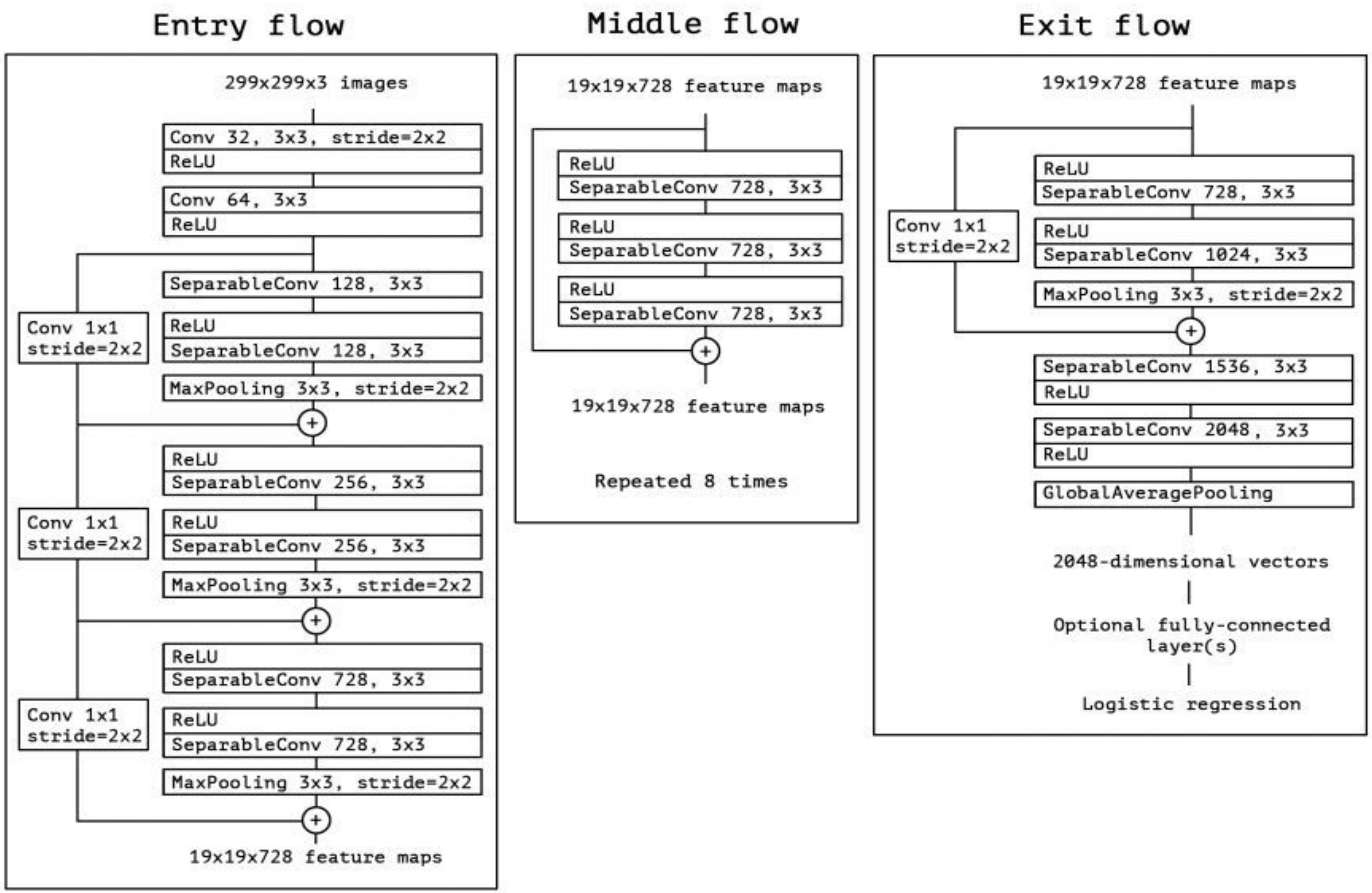


TRANSFER LEARNING

WHAT MODEL HAVE WE USED?

This method takes a model trained on a large dataset and *transfers* its knowledge to a smaller dataset. We exploit a pre-trained neural network and added new layers specific to our problem.

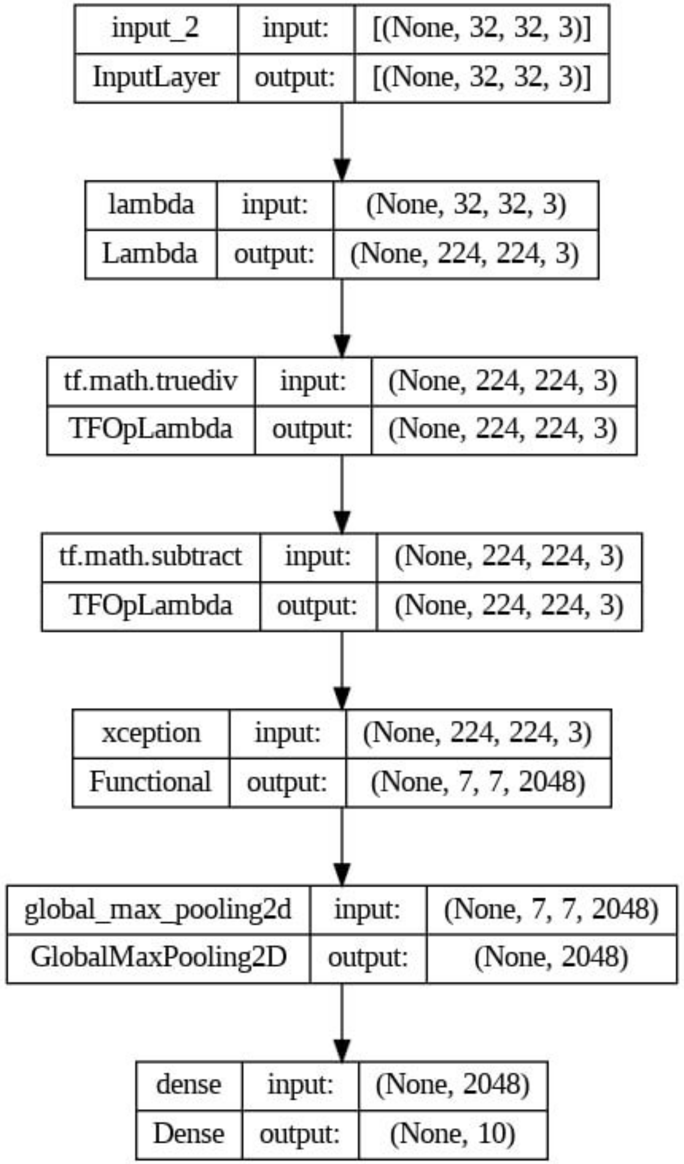
We used the **Xception** network, that achieves 94.5% top-5 accuracy on ImageNet, which is a dataset with over 14 million images belonging to 1000 classes. The graph it's shown the architecture of this network.



TRANSFER LEARNING

WHAT HAS BEEN ADDED?

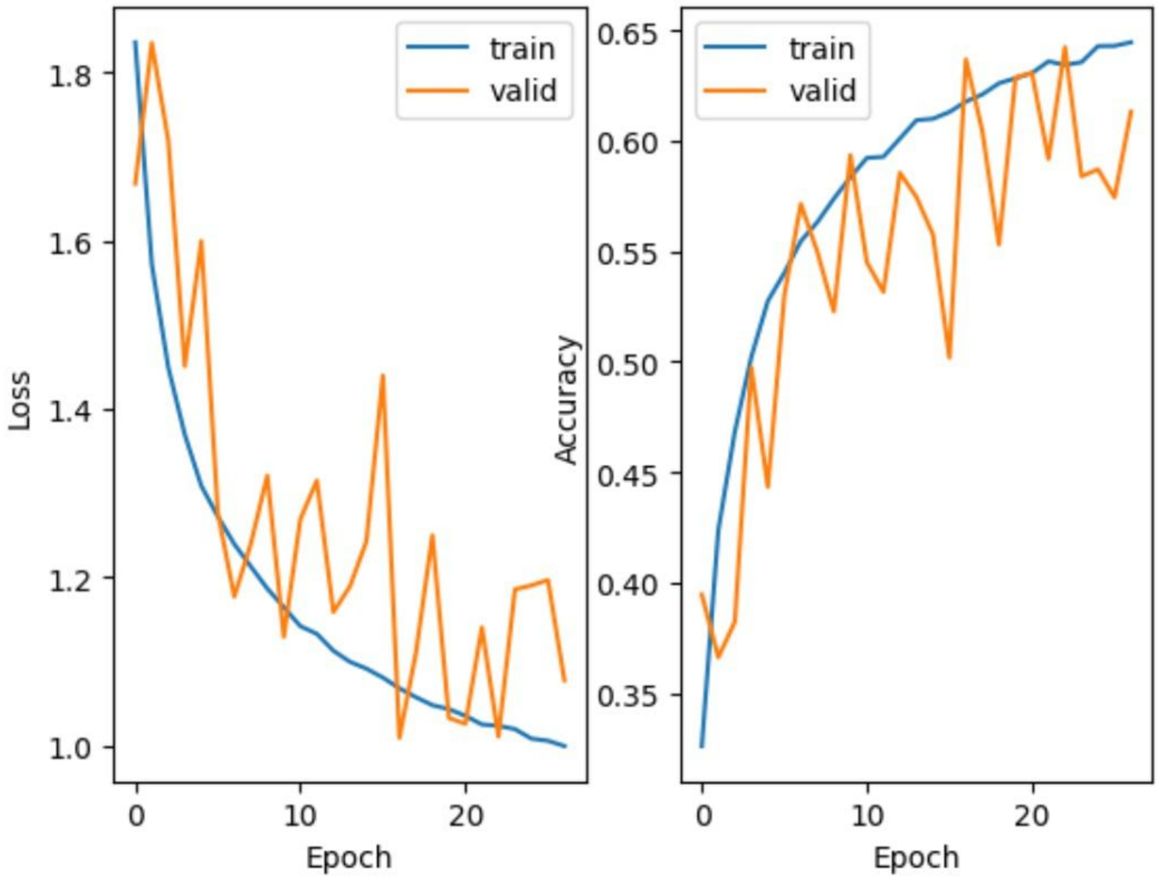
We added a **GlobalMaxPooling2D layer** and a **Dense layer** to the pre-trained model.
After 3 epochs, an accuracy of *0.8648* was achieved.
As expected, using transfer learning we are able to achieve better results with less expensive computation.



1° FAILED ATTEMPT

WHY NOT?

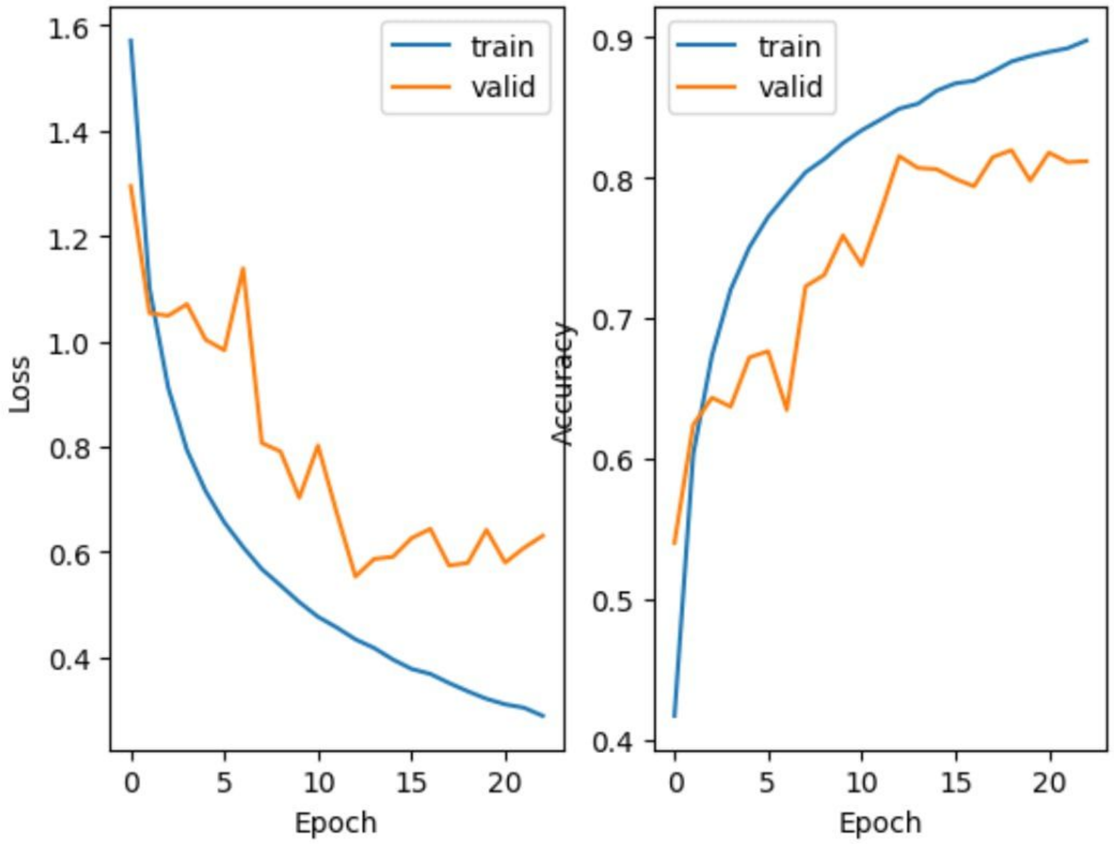
In this first failed attempt we expanded our 2nd model adding another Convolutional layer with *256* filters.
From the graph it can be seen that the result is not satisfactory as there is a worsening in the validation accuracy and an increase in overfitting, giving the fact that the training has been stopped before reaching the 30th epoch.



2° FAILED ATTEMPT

WHY NOT?

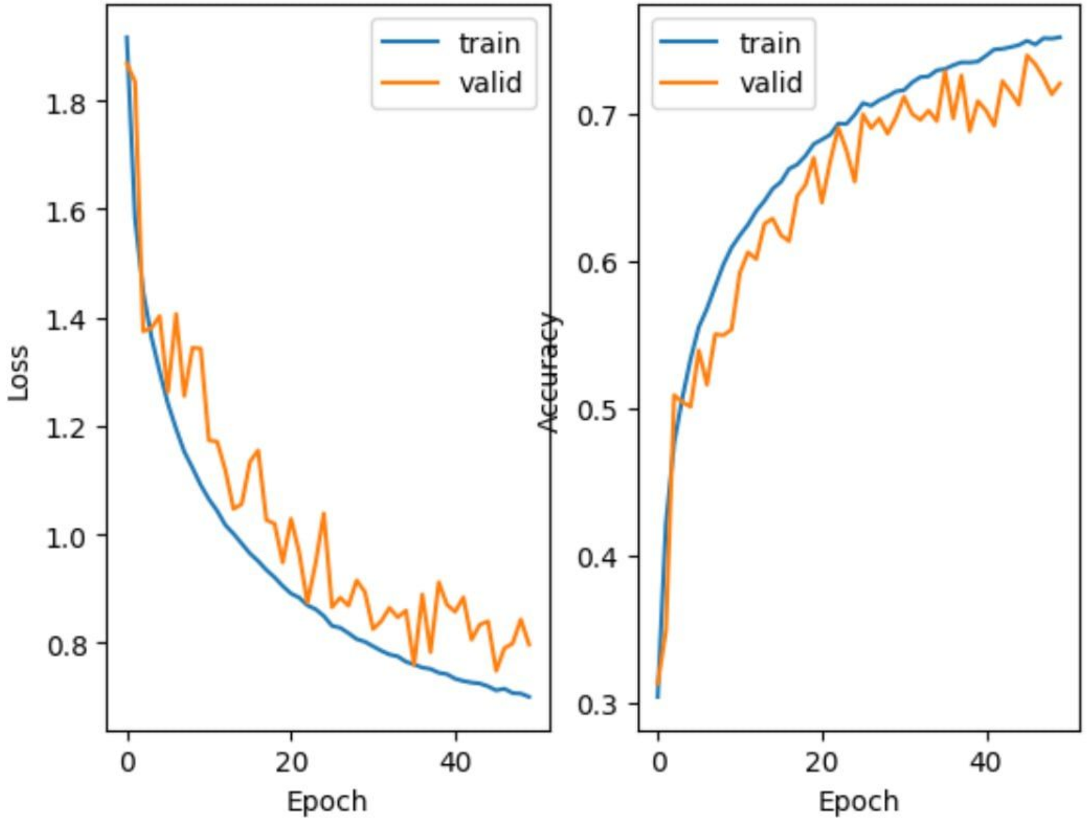
In this second attempt we implemented our 3rd model but with a different kernel size equal to 5 in each Conv2D layer.
Also in this case there is a clear increase in overfitting.



3° FAILED ATTEMPT

WHY NOT?

In this last attempt we used data augmentation techniques trying to reduce overfitting on our 3rd model.
This way we have successfully reduced overfitting, but we have also obtained a lower accuracy on the validation set.



CONCLUSIONS

- The developed CNN architecture has satisfied our research aims, reaching an accuracy on the validation set of ~ 84%.
- The model is better at capturing the features of vehicles rather than the ones of animals (accuracy on the formers not lower than 83%, accuracy on the latters reaching a lowest of 65%).
- Despite having tried to adopt data augmentation techniques, we couldn't achieve a better accuracy; possibly because of the small resolution of the images.
- Surely the model can be upgraded choosing other techniques or parameters to optimize the performance: the comparison with the pre-trained model shows that there is a margin of improvement on the model built from scratch. However, this margin is not substantial (2% difference).

THANK YOU FOR YOUR ATTENTION!