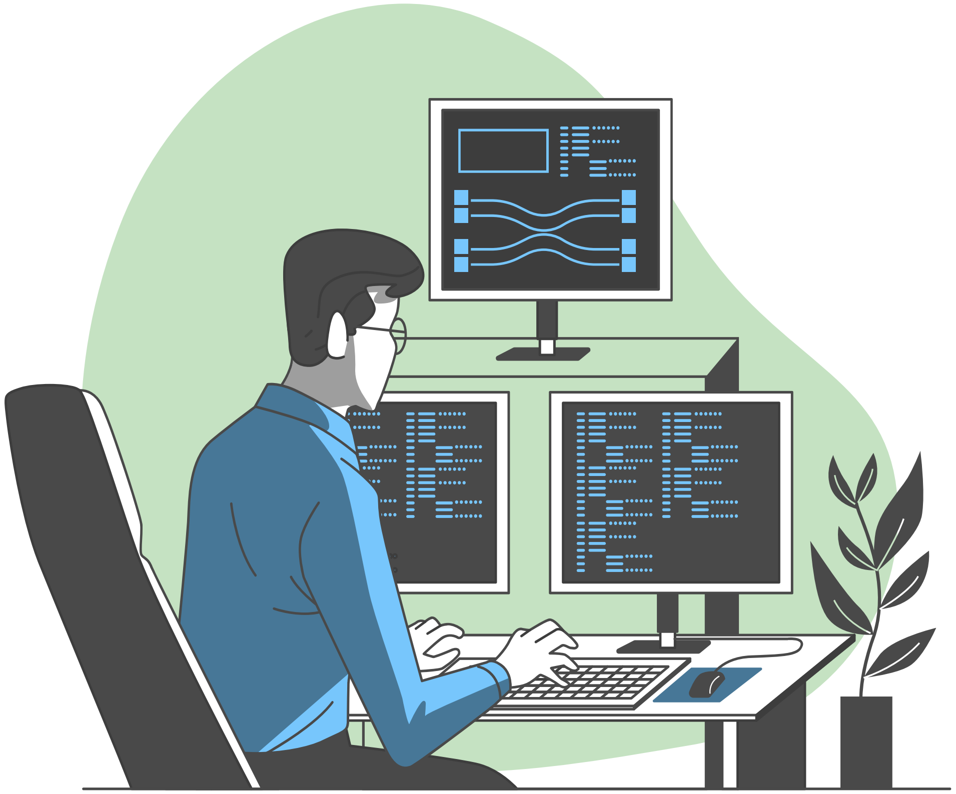


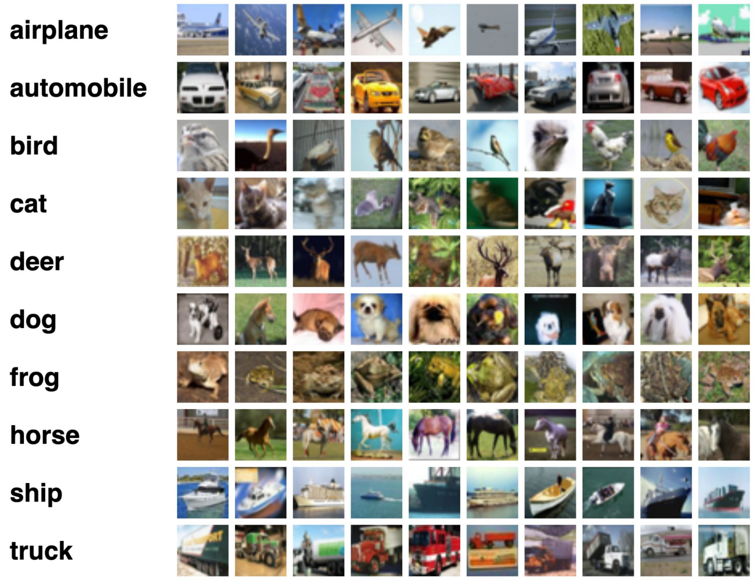
CLASSIFICATION OF CIFAR-10

Foundations of Deep Learning
2022/2023

...

Team:
- **Emanuela Elli** (892901)
- **Federica Madon** (825628)

PURPOSE OF THE PROJECT AND THE DATASET



Purpose of the project: CIFAR-10 dataset is a collection of common images, the model must be able to correctly recognize the type of object or animal represented. Then it has to perform a *classification task*.

The considered dataset is composed by **60000 color images** (32x32, 3 colors RGB) divided in 10 classes (*airplane*, *automobile*, *bird*, *cat*, *deer*, *dog*, *frog*, *horse*, *ship*, *truck*). Each image has a single label.

All these classes are mutually exclusive. Each class is formed by 6000 images.

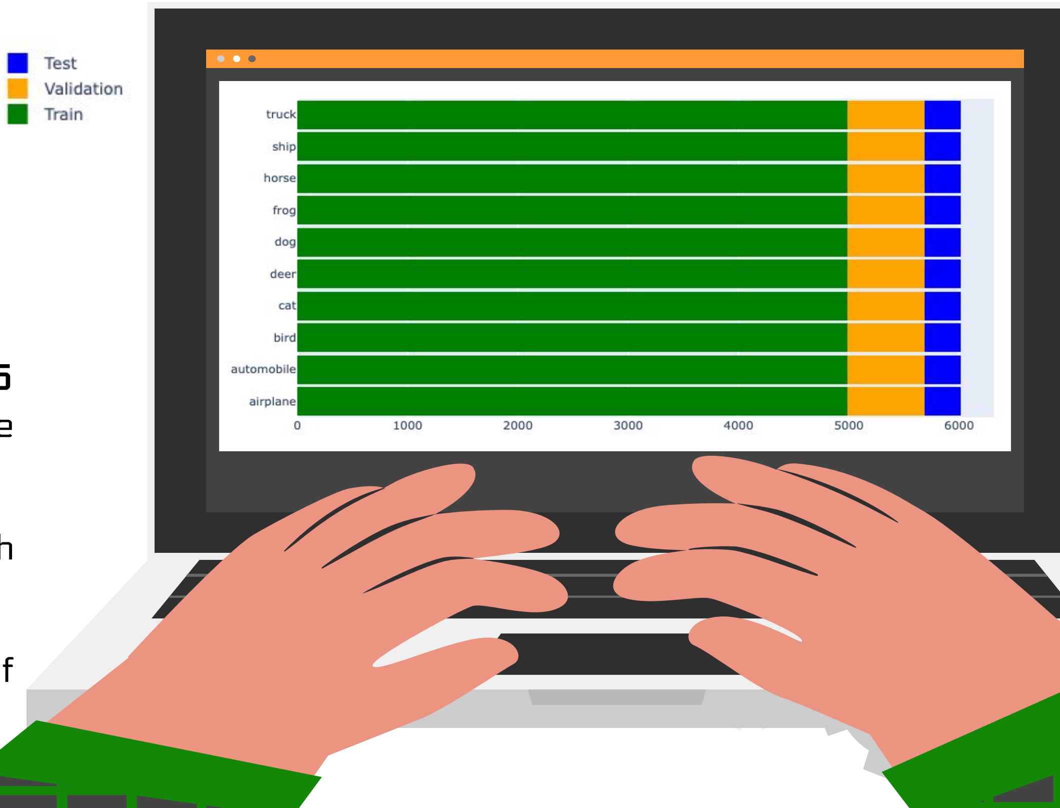
DATA MANAGEMENT AND PRE-PROCESSING

The dataset is composed of:

- 50000 images in the training set;
- 10000 images in the test set.

The test set is divided in:

- 7000 images for the validation set;
- 3000 images for the test set.



Normalization and One-Hot Encoding

- All the pixels of images have values in **0-255** range. It's useful, for the efficiency of the model, to normalize these values in **0-1** range.
- The labels of images are encoded with **One-Hot Encoding**.
- Each set of data has the same number of images for each class.

SOLUTION OF THE PROBLEM

For this task a **CNN** has been chosen.

Parameters

- **50** epochs
- Loss = **Categorical-Crossentropy**
- **ReLu** and **Softmax** function
- Different **optimizers**

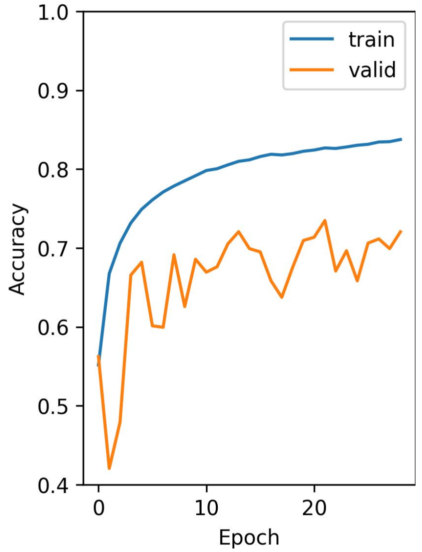
Three Networks

Three different **networks** are developed with different layers and regularizations.

Early stopping is defined with *patience = 15* and the loss of the validation set is monitored.

Model 1

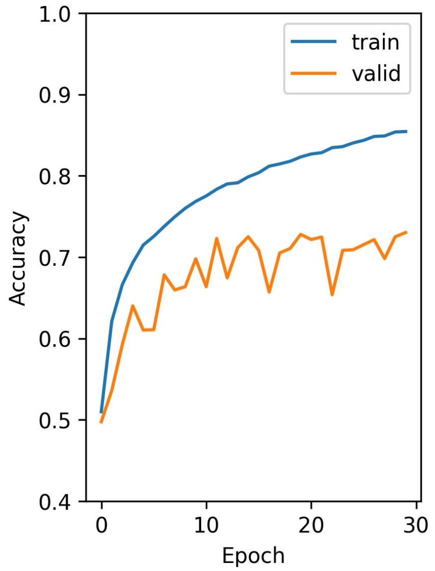
Loss: 0.76
Validation Loss: 1.13



Three Conv2D blocks and two fully connected layers.

Model 2

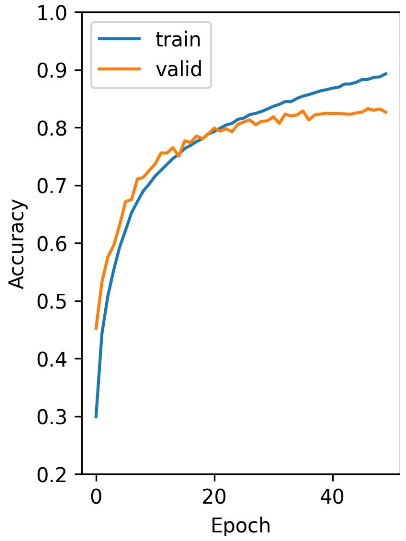
Loss: 0.42
Validation Loss: 0.89



Six coupled Conv2D layers and two fully connected layers.

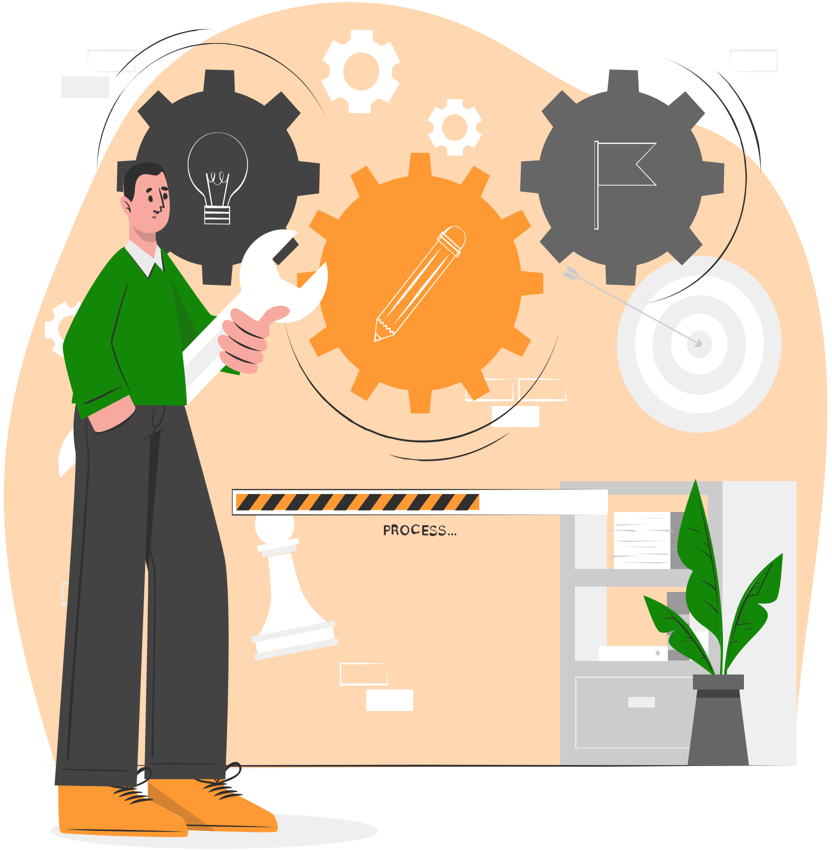
Model 3

Loss: 0.49
Validation Loss: 0.57



Six coupled Conv2D layers and two fully connected layers

MODEL CHOSEN AND IMPROVEMENTS



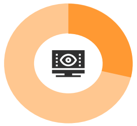
Model 3

Optimizer = SGD
Weight initializer = he uniform
Dropout

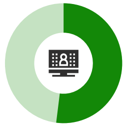
Improvements

Data Augmentation
Batch Normalization
Increasing Dropout
More Epochs

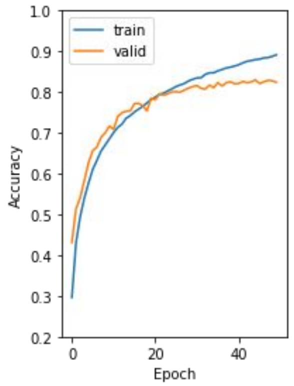
MODEL CHOSEN

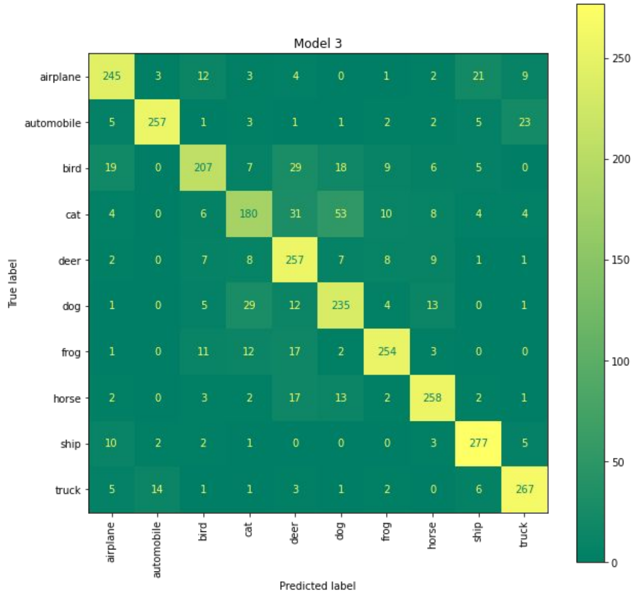


0.49
Train Loss



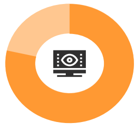
0.57
Validation Loss



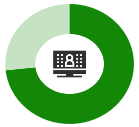


	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	245	3	12	3	4	0	1	2	21	9
automobile	5	257	1	3	1	1	2	2	5	23
bird	19	0	207	7	29	18	9	6	5	0
cat	4	0	6	180	31	53	10	8	4	4
deer	2	0	7	8	257	7	8	9	1	1
dog	1	0	5	29	12	235	4	13	0	1
frog	1	0	11	12	17	2	254	3	0	0
horse	2	0	3	2	17	13	2	258	2	1
ship	10	2	2	1	0	0	0	3	277	5
truck	5	14	1	1	3	1	2	0	6	267

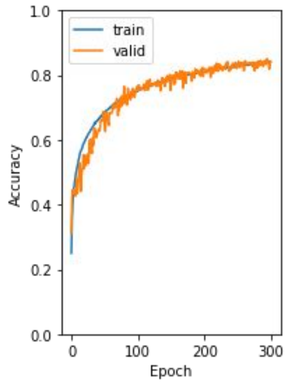
IMPROVEMENTS

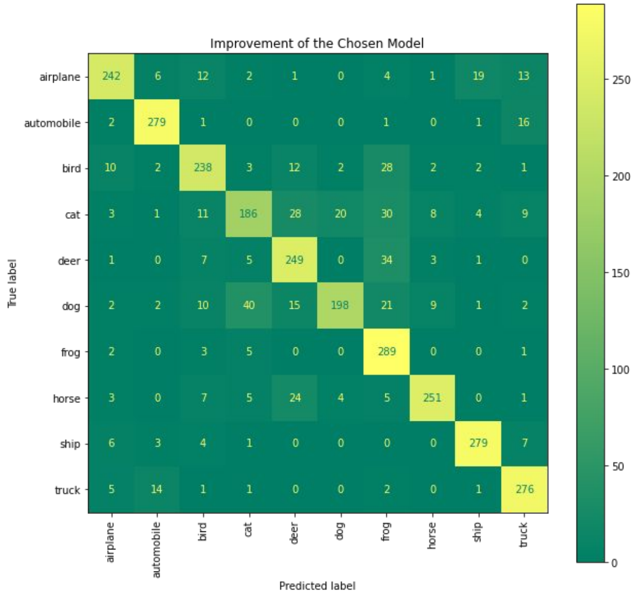


0.46
Train Loss



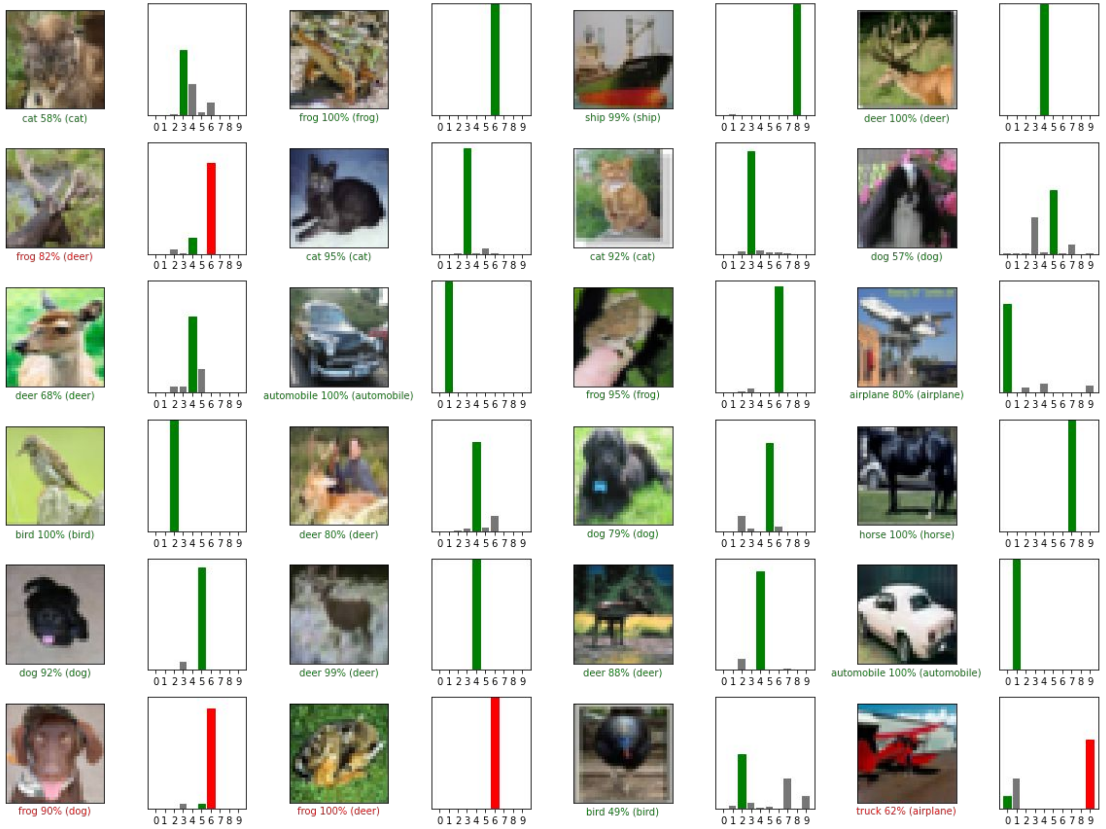
0.48
Validation Loss

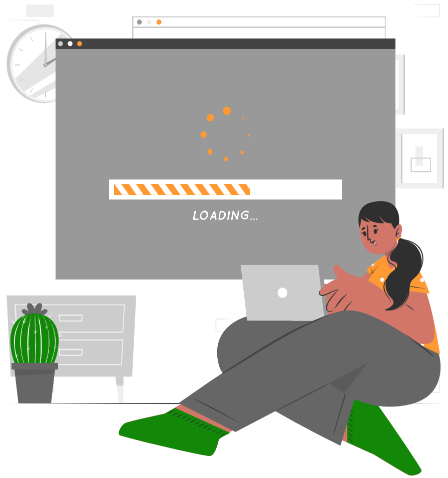




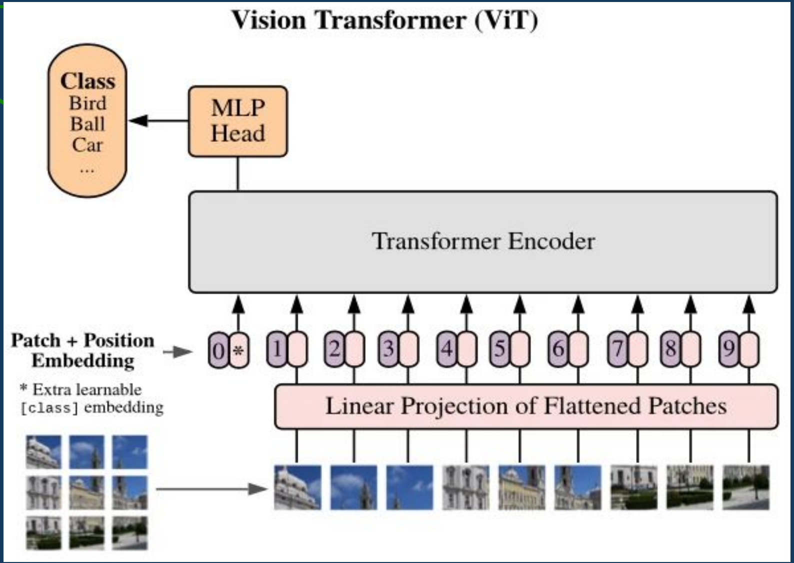
	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	242	6	12	2	1	0	4	1	19	13
automobile	2	279	1	0	0	0	1	0	1	16
bird	10	2	238	3	12	2	28	2	2	1
cat	3	1	11	186	28	20	30	8	4	9
deer	1	0	7	5	249	0	34	3	1	0
dog	2	2	10	40	15	198	21	9	1	2
frog	2	0	3	5	0	0	289	0	0	1
horse	3	0	7	5	24	4	5	251	0	1
ship	6	3	4	1	0	0	0	0	279	7
truck	5	14	1	1	0	0	2	0	1	276

TEST ON FEW IMAGES OF THE FINAL MODEL





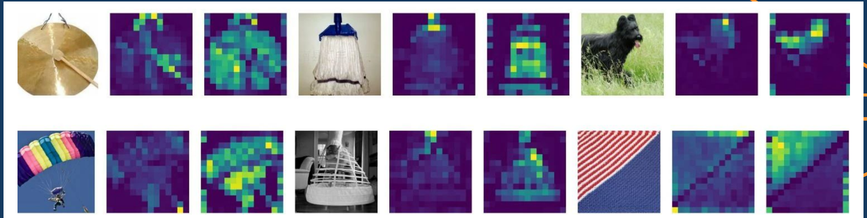
ViT - Vision Transformers


Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 *
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

In 2022, the Vision Transformer (ViT) emerged as a competitive alternative to convolutional neural networks (CNNs) that are currently state-of-art in computer vision and therefore widely used in different image recognition tasks.

The ViT is a visual model based on the architecture of a transformer originally designed for text-based tasks. The ViT model represents an input image as a series of image patches (like the series of word embeddings used by transformers for text classification) and it predicts directly class labels for the image.

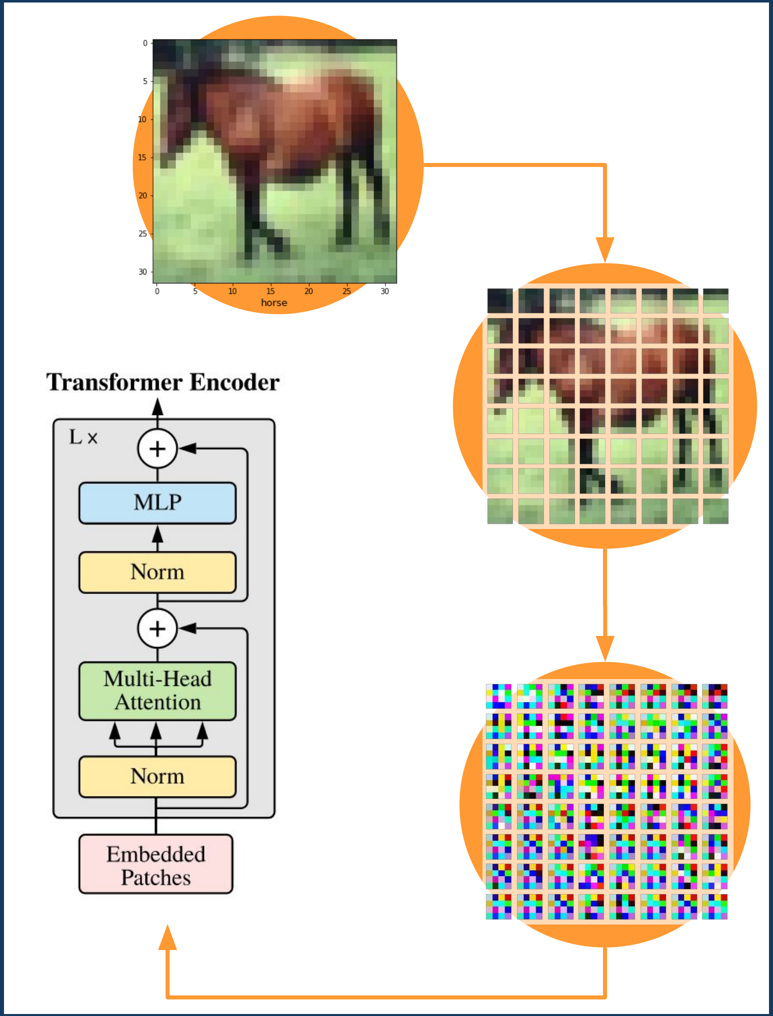
CNN uses pixel arrays, whereas ViT splits the images into visual tokens. The visual transformer divides an image into fixed-size patches, correctly embeds each of them, and includes positional embedding as an input to the transformer encoder.


Raw images (left) with attention maps of the ViT-S/16 model

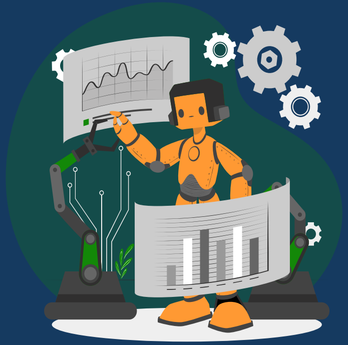
ViT Architecture

The overall architecture of the vision transformer model is given by following these step-by-step manners:

- Split an image into **patches** (fixed sizes);
- **Flatten** the image patches;
- Create lower-dimensional linear embeddings from these flattened image patches (we can think of these now as "**tokens**");
- Include **positional embeddings**;
- Feed the sequence as an input to a state-of-art **transformer encoder**;
- Fine-tune the dataset for **image classification**.

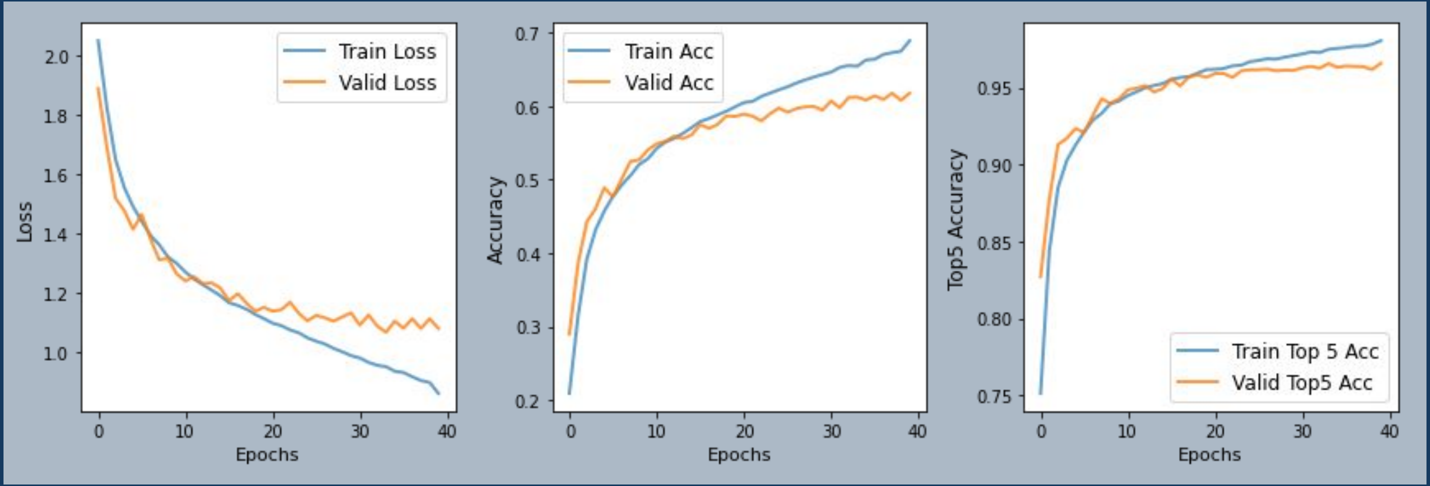


ViT Outcome

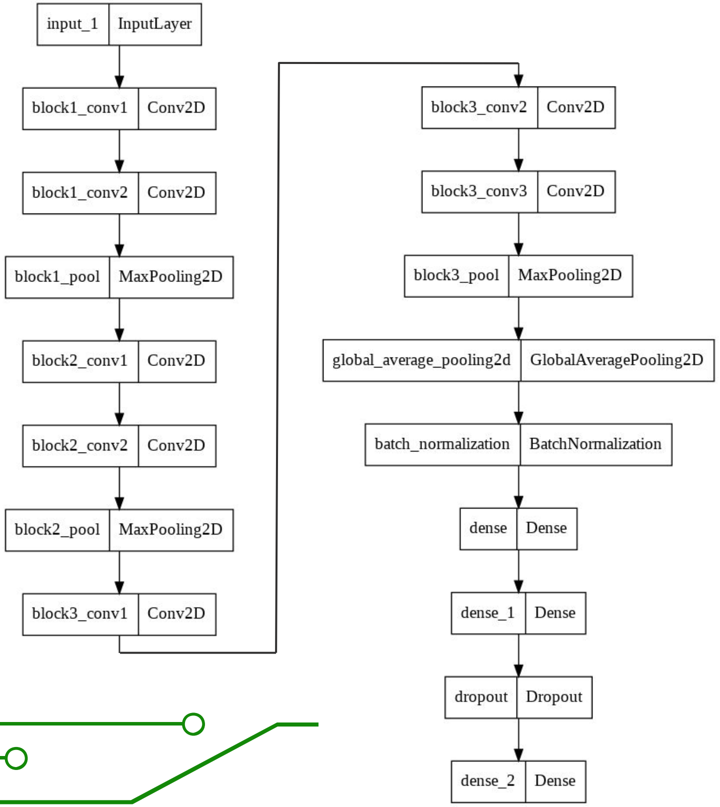


Accuracy on the train set of ViT model: 0.69
Accuracy on the validation set of ViT model: 0.62

Loss on the train set of ViT model: 0.86
Loss on the validation set of ViT model: 1.08

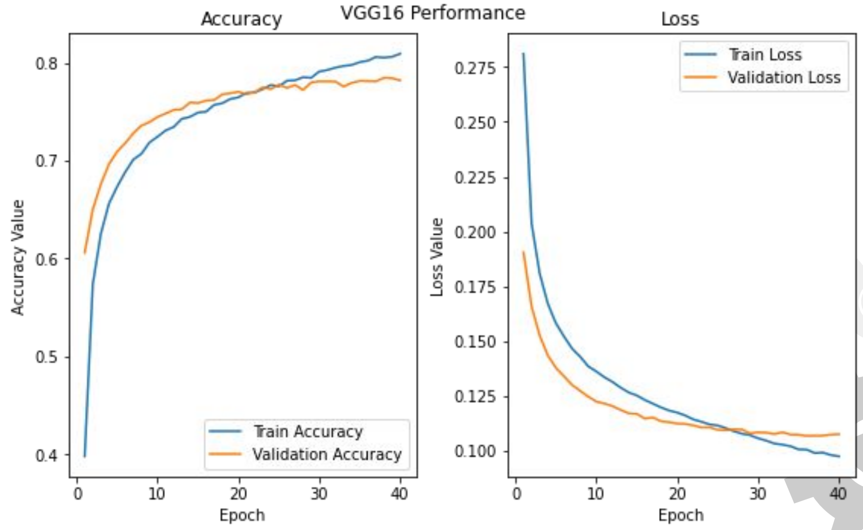


TRANSFER LEARNING: VGG16

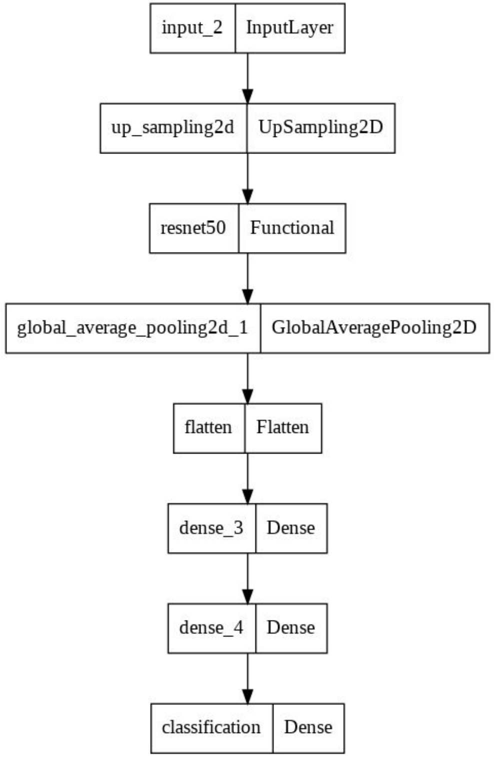


```
Accuracy on the train set of VGG16: 0.81
Accuracy on the validation set of VGG16: 0.78

Loss on the train set of VGG16: 0.10
Loss on the validation set of VGG16: 0.11
```

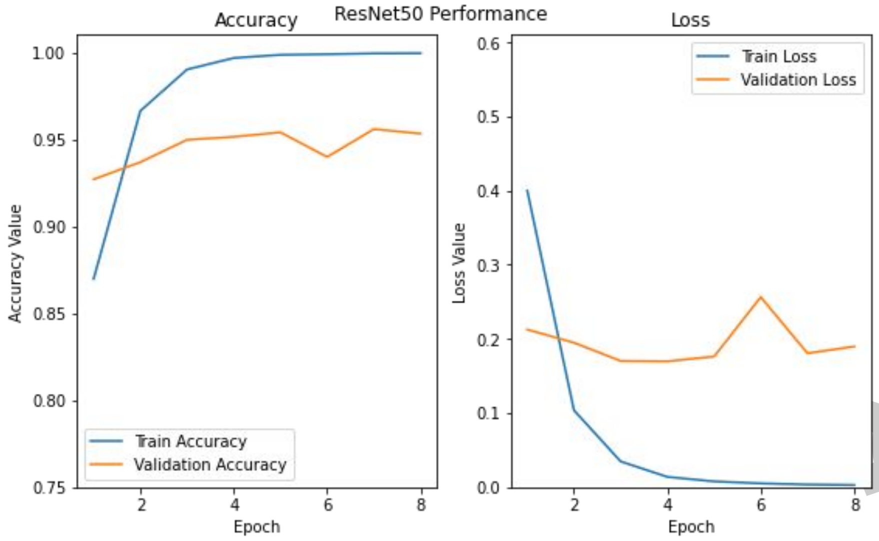


TRANSFER LEARNING: ResNet50



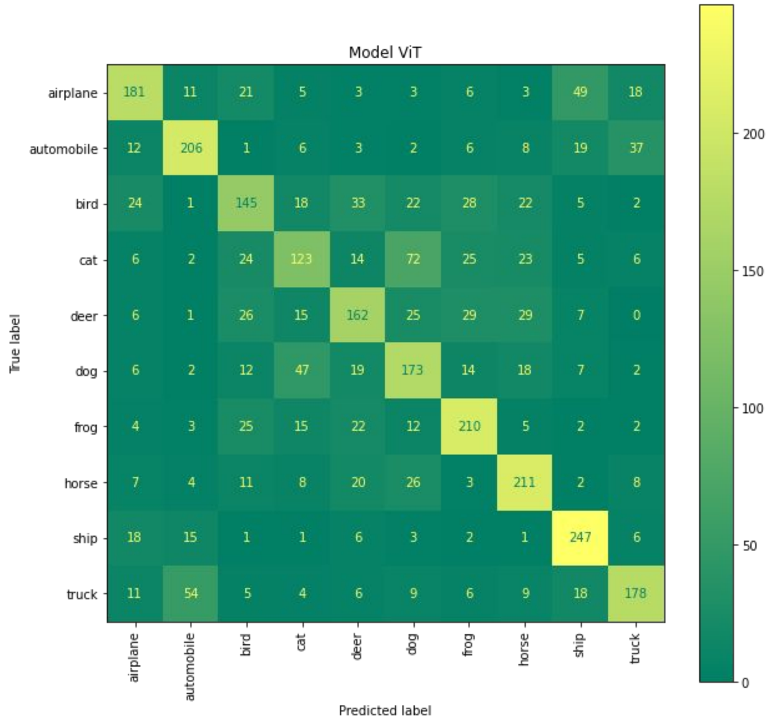
```
Accuracy on the train set of ResNet50: 0.99
Accuracy on the validation set of ResNet50: 0.95

Loss on the train set of ResNet50: 0.03
Loss on the validation set of ResNet50: 0.20
```



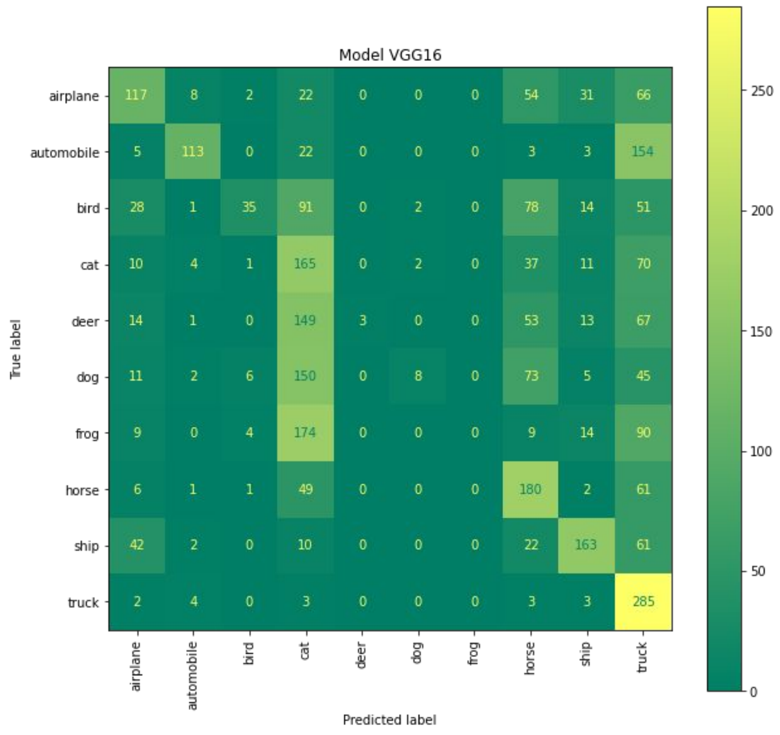
OUTCOME AND FUTURE DEVELOPMENT

ViT



True label \ Predicted label	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	181	11	21	5	3	3	6	3	49	18
automobile	12	206	1	6	3	2	6	8	19	37
bird	24	1	145	18	33	22	28	22	5	2
cat	6	2	24	123	14	72	25	23	5	6
deer	6	1	26	15	162	25	29	29	7	0
dog	6	2	12	47	19	173	14	18	7	2
frog	4	3	25	15	22	12	210	5	2	2
horse	7	4	11	8	20	26	3	211	2	8
ship	18	15	1	1	6	3	2	1	247	6
truck	11	54	5	4	6	9	6	9	18	178

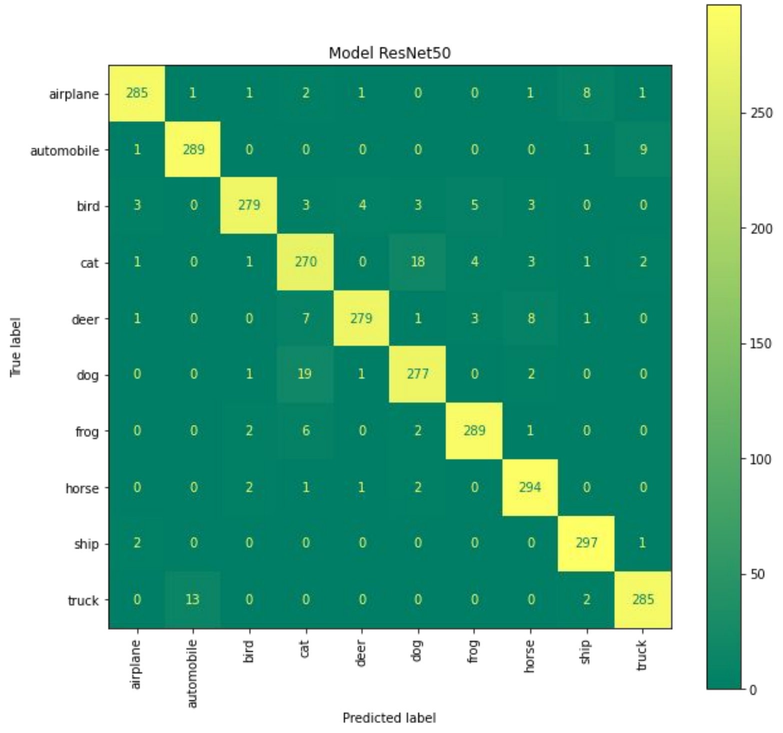
VGG16



True label \ Predicted label	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	117	8	2	22	0	0	0	54	31	66
automobile	5	113	0	22	0	0	0	3	3	154
bird	28	1	35	91	0	2	0	78	14	51
cat	10	4	1	165	0	2	0	37	11	70
deer	14	1	0	149	3	0	0	53	13	67
dog	11	2	6	150	0	8	0	73	5	45
frog	9	0	4	174	0	0	0	9	14	90
horse	6	1	1	49	0	0	0	180	2	61
ship	42	2	0	10	0	0	0	22	163	61
truck	2	4	0	3	0	0	0	3	3	285

OUTCOME AND FUTURE DEVELOPMENT

ResNet50

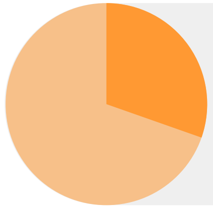


True label \ Predicted label	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	285	1	1	2	1	0	0	1	8	1
automobile	1	289	0	0	0	0	0	0	1	9
bird	3	0	279	3	4	3	5	3	0	0
cat	1	0	1	270	0	18	4	3	1	2
deer	1	0	0	7	279	1	3	8	1	0
dog	0	0	1	19	1	277	0	2	0	0
frog	0	0	2	6	0	2	289	1	0	0
horse	0	0	2	1	1	2	0	294	0	0
ship	2	0	0	0	0	0	0	0	297	1
truck	0	13	0	0	0	0	0	0	2	285

For the model chosen we suggest to:

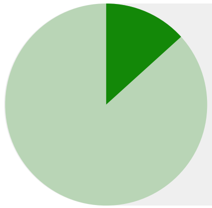
- ➔ Change **learning rates** (optimizer);
- ➔ **Resize pixels**, e.g. with standardization;
- ➔ Use different **regularization techniques** (e.g. increase dropout, etc);
- ➔ Explore other existing **more efficient** networks;
- ➔ Try a **data ensemble** approach;
- ➔ Use **PCA** to reduce noise in data (as shown in the next slide).

FUTURE DEVELOPMENTS - PCA



29%

Variance captured from the first component.

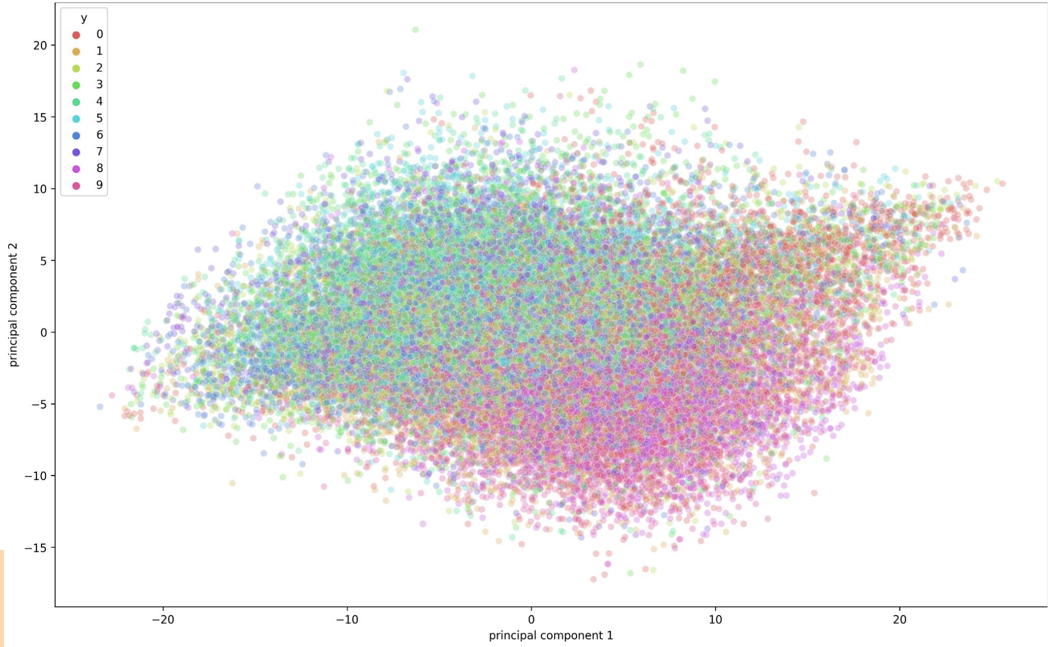


11%

Variance captured from the second component.

Points or images belonging to the same class are closed to each other.

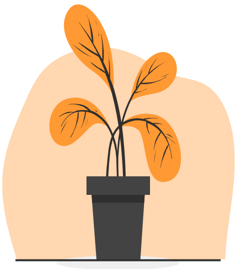
Points or images that are very different semantically are farther from each other.



RESOURCES

RELEVANT WEBSITES:

- *datacamp.com/tutorial/principal-component-analysis-in-python*
- *geeksforgeeks.org/cifar-10-image-classification-in-tensorflow/*
- *pythonistaplanet.com/cifar-10-image-classification-using-keras/*
- *machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-cifar-10*
- *kaggle.com/code/faressayah/cifar-10-images-classification-using-cnns-88*
- *towardsdatascience.com/understand-and-implement-vision-transformer*
- *viso.ai/vision-transformer-vit/The_vision_transformer_model_uses,processed_by_the_transformer_encoder*
- *github.com/sayakpaul/Transfer-Learning-with-CIFAR10/VGG16_Classifier.ipynb*
- *kaggle.com/resnet50-transfer-learning-cifar-10*



Thank you for your attention!