CONTENTS

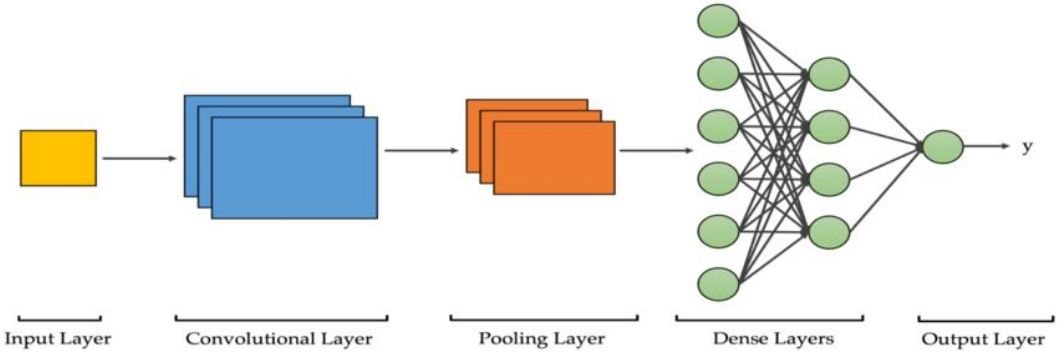
- Description of a problem
- An analysis of a problem
- Analysis of the dataset
- Some existing solutions
- Description of a preferred solution
- Code implementation
- Simulation of an application
- Conclusions

DESCRIPTION OF A PROBLEM

The concept of the project is to provide a sign recognition system.

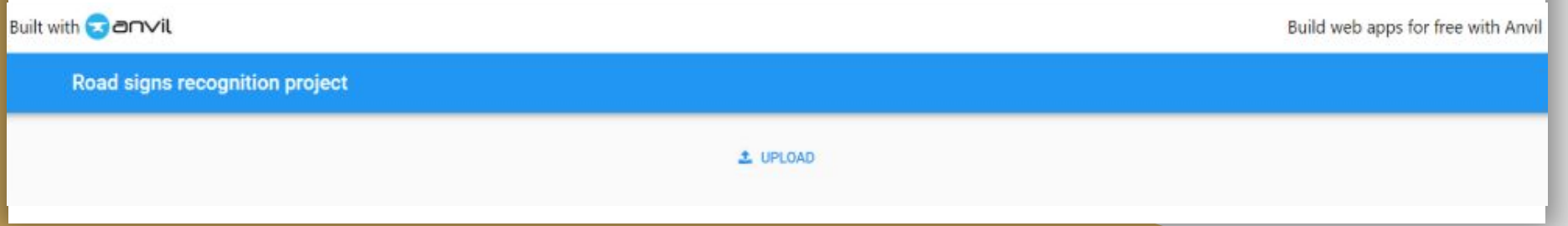
The **input** will be an image which it will be categorized using a convolutional neural network (CNN).

The **output** will be in a form of text with a description of a sign and a formatted part of an image with the sign.



AN ANALYSIS OF A PROBLEM

- The application send the image to the **Google Drive folder**.
- Then the program will give us the answer of what **type of sign** is it.
- We will check accuracy by the test set with around **12000** pictures.
- We will mainly use **Keras's, tensorflow** and some libraries to maintain a large set of data.
- The main problem is to set a proper type of model - we have chosen sequential one.



ANALYSIS OF THE DATASET

- The whole dataset contains over 53000 images.
- It is divided between test and train sets.
  - Test set is around 40000 images in 43 different classes. .
- To properly load and test the dataset we should divide the train set into two subsets - test and train.

SOME EXISTING SOLUTIONS

1. Classify the different types of signals: using Spatial Transformer Networks, with a modified version of IDSIA networks.

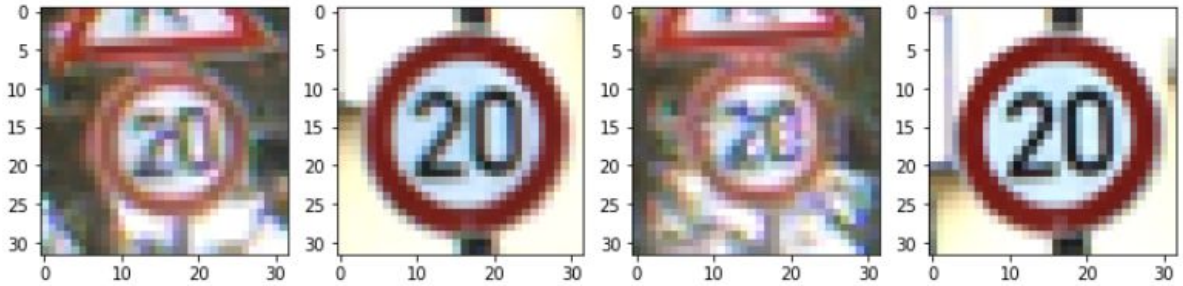
2. Identifying the existence of a sign in the photo.

3. Detecting multiple signs.

DESCRIPTION OF A PREFERRED SOLUTION

Our application will have two main parts:

- Connected with Artificial Intelligence:
  - Analyzing the dataset and transformation of data.
  - The plan is to save pictures as 32x32 pixels objects.



- Next we are going to develop a Convolutional Neural Network model with 42 classifiers. Then the model will be fitted.

```
Epoch 1/4
491/491 [==============================]
Epoch 2/4
491/491 [==============================]
Epoch 3/4
491/491 [==============================]
Epoch 4/4
491/491 [==============================]
```

- Connected to Graphical User Interface:
  - We will develop a simple view with an option of uploading an image and fit it into the model created before.

Code implementation

First, a preprocessing layer which **rescales** input values to a new range.

Then we create a **convolution kernel** to produce a tensor of outputs.

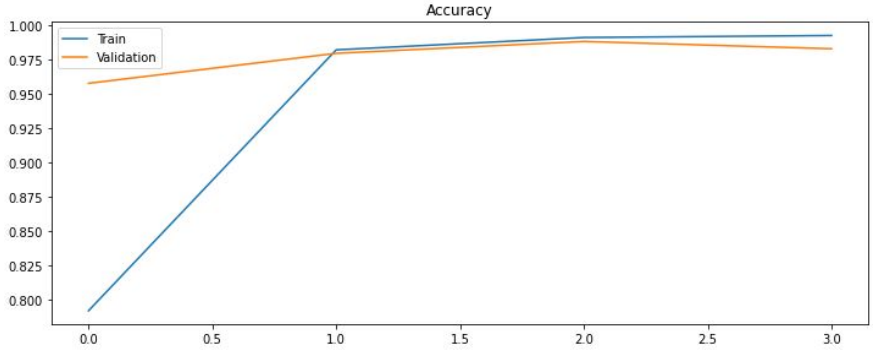
By taking the **maximum** value over an input window for each channel of the input.

And we are going to repeat this processes to train our model

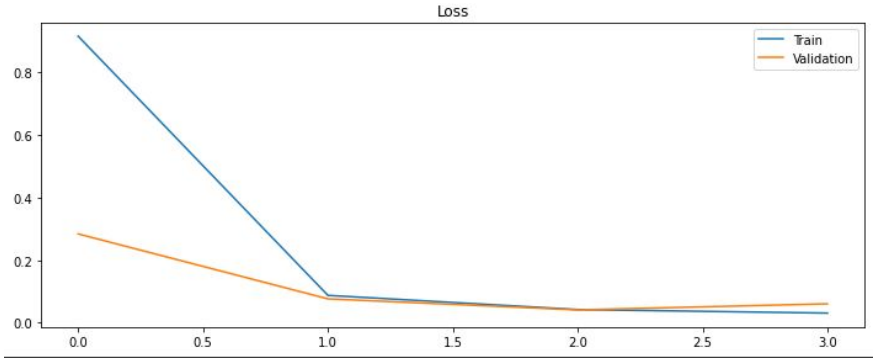
```
from tensorflow.keras import models, layers

model = keras.models.Sequential([
    keras.layers.Rescaling(scale=1./255.),
    keras.layers.Conv2D(filters=72, kernel_size=(3, 3), strides=1,
              activation="relu",  input_shape=X.shape[1:]),
    keras.layers.MaxPooling2D(pool_size=(2), strides=2),
    keras.layers.Conv2D(filters=72, kernel_size=(5, 5), strides=1, activation="relu"),
    keras.layers.MaxPooling2D(pool_size=(2), strides=2),
    keras.layers.Conv2D(filters=144, kernel_size=(5, 5), activation="relu"),
    keras.layers.MaxPooling2D(pool_size=(1)),
    keras.layers.Flatten(),
    keras.layers.Dense(units=86, activation="relu"),
    keras.layers.BatchNormalization(),
    keras.layers.Dropout(0.2),
    keras.layers.Dense(units=43, activation="softmax"),
  ])
```

Training evaluation

- The **higher** the training, the better the **accuracy**. So the model is able to **identify more signals correctly** among all possible signals.

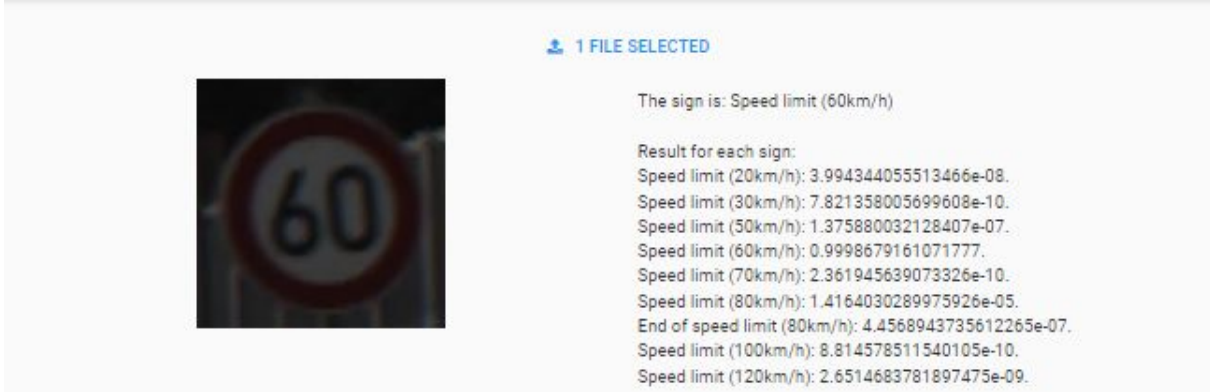


- Estimates the **desviation** between the **predictions** made by the neural network and the **actual values** of the observations used during learning



SIMULATION OF AN APPLICATION

- The website is build using **Anvil**.
- The UI is pretty simple, user can choose the **image.**
- **O**ur program will detect the **type of the sign**, or if there is no sign, user will get the corresponding message.



CONCLUSIONS

This project will find its usefulness in improving the automatic cars, and it will bring more safety to the art of driving a vehicle.

The app could be improved with more complex functions. For example, identify the signal even if the image is not of good quality, or if the background of the image complicates the identification of the signal…