# 

Warsaw University of Technology

# 

# 

# 

# **[EARIN] Project: Facial keypoint detection**

## **Matic Absec, Kibirima Kiatihou Hienrich**

Summer 2023

Authors: Matic Absec ([matic.absec.stud@pw.edu.pl](mailto:matic.absec.stud@pw.edu.pl)),

Hienrich Kibirima Kiatihou ([hienrich.kibirima\_kiatihou.stud@pw.edu.pl](mailto:hienrich.kibirima_kiatihou.stud@pw.edu.pl))

**PRELIMINARY DOCUMENTATION**

This code is a facial key points detection model. The model is based on a convolutional neural network and uses the Mean Squared Logarithmic Error as the loss function.

The input data is a set of 96x96 grayscale images, where the target is to detect some key points on the face, such as the location of the eyes, nose, and mouth. The model has 7 convolutional layers and 2 dense layers, with ReLU as the activation function for the convolutional layers and linear for the output layer.

The model is trained on the training set and the performance is evaluated on the mean squared logarithmic error metric. Finally, the model is used to predict the key points on the test set.

**1- Algorithms:**

A deep learning algorithm based on convolutional neural networks (CNN) is used for the detection of key points of the face. A sequential model was defined using the Keras library and included several layers of convolution, density and pooling operations. The logarithmic mean square error was used as the loss function and the Adam optimizer was used to train the model.

**2- Datasets:**

Used the facial keypoint detection dataset available on Kaggle, which includes a training set with 7049 images and the corresponding 30 keypoint coordinates, a test set with 1783 images, and an IdLookupTable to map the predicted keypoints to facial features. The image and label data were extracted from the CSV files, preprocessed by filling in missing values with the previous value and reshaped into a 96x96 greyscale image format.

**3- General test/experimental design:**

Trained the model on the training set using a batch size of 128 and 60 epochs. Evaluated the performance of the model using log-mean-square error as the loss metric and log-mean-square error as the evaluation metric. Predictions were made on the test set and saved.

**4- Methods for visualizing the results:**

We used Matplotlib to display a sample image from the training and test sets, and visualised the predicted coordinates of the key points.

**5- Quality Measures:**

We used the mean squared logarithmic error as the loss function and the mean squared logarithmic error as the evaluation metric to assess the performance of the model.