Facial Keypoint Detection

Objective: The objective is to detect facial keypoint positions on face images from the given facial dataset.

Image Keypoints:

- Image keypoints are distinct points or places in an image that are picked because they represent unique and distinguishable features.
- These points are crucial in computer vision and image processing for various tasks such as image recognition, object detection, and image matching.

Problem categorization:

- Keypoint detection is a **regression task** where output is a continuous value of predicted keypoints. The input images are and the corresponding keypoints are the input to the network and the output is the keypoints predicted by the model.

Dataset:

- Each keypoint is specified by an (x,y) real-valued pair in the space of pixel indices. There are 15 keypoints in the dataset, which represent the following elements of the face:
- Left_eye_center, right_eye_center, left_eye_inner_corner, left_eye_outer_corner, right_eye_inner_corner, right_eye_outer_corner, left_eyebrow_inner_end, left_eyebrow_outer_end, right_eyebrow_inner_end, right_eyebrow_outer_end, nose_tip, mouth_left_corner, mouth_right_corner, mouth_center_top_lip, mouth_center_bottom_lip

Dataset_link:
https://www.kaggle.com/datasets/nagasai524/facial-keypoint-detection/data

Challenges in the Facial Keypoint detection:

- Detecting facial keypoints is a very challenging problem. Facial features vary greatly from one individual to another, and even for a single individual, there is a large amount of variation due to 3D pose, size, position, viewing angle, and illumination conditions.

Project folder structure organization:

- **Project structure:**

```
keypoint_detection/
      training.csv
      kaggle_facial_keypoint_dataset.zip
      images/
          train_images/
           - 1.jpg
           - 2.jpg …
          test_images/
           - 1.jpg
           - 2.jpg …
          Processed_keypoint_data.npz
      model_results/
           base_model/
                 base_model_summary.txt
                 base_model_plot.png
                 training_validation_loss_plot.png
                 trained_base_model_keypoint_detection.h5
                 keypoint_predictions.csv
                 base_model_results.png
                 base_model_results_1.png
                 evaluation_results_base_model_keypoint.csv
           model_v1/
                 model_v1_summary.txt
                 model_v1_plot.png
                 model_v1_results.png
                 model_v1_results_1.png
                 evaluation_results_model_v1_keypoint.csv
                 training_validation_loss_plot.png
                 trained_model_v1_keypoint_detection.h5
                 model_v1_keypoint_predictions.csv
      processed_training_labels.csv
      data_analysis_keypoint_detection.ipynb
      Readme_keypoint_detection_document.gdoc
      Readme_keypoint_detection_document.pdf
      base_model_keypoint_detection_script.ipynb
      model_v1_ keypoint_detection_script.ipynb
```

- **Steps followed in the development of the project:**
  - Data analysis for understanding the nature of images and keypoints in the dataset, data preprocessing such as filling in missing keypoints. **(Notebook: data_analysis_yeypoint_detection.ipynb)**
  - The dataset validation after preprocessing.

- Data Loading and converting dataset into numpy array for easy loading and processing while training
- Loading numpy array and splitting the dataset into train, validation, and test (Total samples = 7049, Train = 5639, Validation = 705, Test = 705)
- Two models were developed:
    - Base_model
    - Model_v1
    - Saved summary, plots, predictions and performance of the models
    - **Model notebook**
        - **base_model_keypoint_detection_script.ipynb**
        - **model_v1_ keypoint_detection_script.ipynb**
- Trained two models separately using the following training configuration
    - Optimizer: Adam
    - Learning rate: 1Xe-5
    - Loss: mean squared error
    - Metric: mean absolute error
    - Training callback: EarlyStopping
    - Batch Size: 32
    - Epochs: 200
- Saved the trained model and plot for losses
- Load the model and Model predictions saved in CSV
- visualization of the ground truth and predictions on the sample images
- Performance evaluation using mean absolute error and accuracy of the model for three threshold values (5, 10, 15) of pixels in percentage
- Saved the evaluation performance in CSV file

Result Analysis:

- Both models are evaluated using the Mean Absolute Error (MAE) and the different threshold pixel values.
- The accuracy of the model varies depending on the pixel threshold.
- The result analysis in Table 1 indicated that the performance of the model_v1 is better than the base model.

Table 1: Evaluation performance of the models

Models	Performance evaluation results			
	MAE	MAE_threshold _pixels_5_in%	MAE_threshold _pixels_10_in%	MAE_threshold _pixels_15_in%
base_model	3.438	77.962	94.676	98.572
model_v1	2.023	92.988	99.177	99.806

- The model parameters and the memory requirements are given below in Table 2.

Table 2: Model parameters

Models	Parameters
base_model	6662750
model_v1	6668378

- Table 1 and Table 2 indicate that the model_v1 is performing better than the base_model. However, the model_v1 has more parameters than the base_model.
- The parameter increases due to the embedding of the squeeze-and-excitation network (SENet) in the architecture.
- The SENet has the ability to capture the significant features from the input feature map and uplift the learning and generalization capability of the model. It acts as channel-wise attention in the network that learns the significant details and suppresses the redundant information. The architecture of the SENet shown in Figure 1 is utilized for the model architecture building.
- The computational complexity of the SENet can be controlled by adjusting the parameter ratio (r) in the model architecture. In the model_v1 r = 8 is chosen for maintaining the complexity and performance. If the value of r-increases the complexity reduces and vice versa.

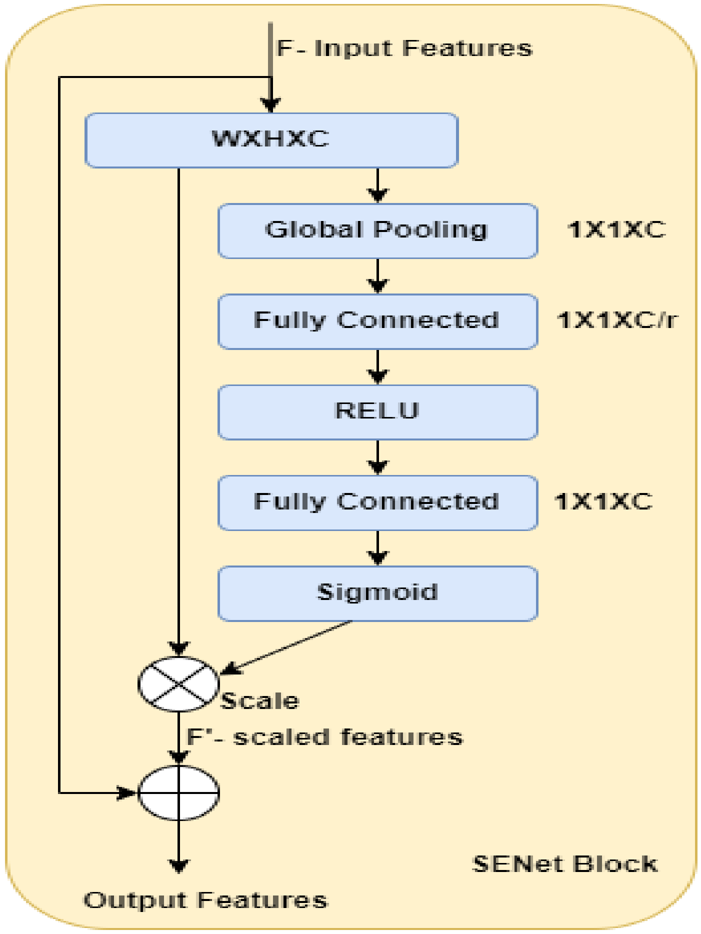

Figure 1: SENet architecture

- The difference in the architecture of both the networks is shown in the following Figure 2 and Figure 3 respectively.

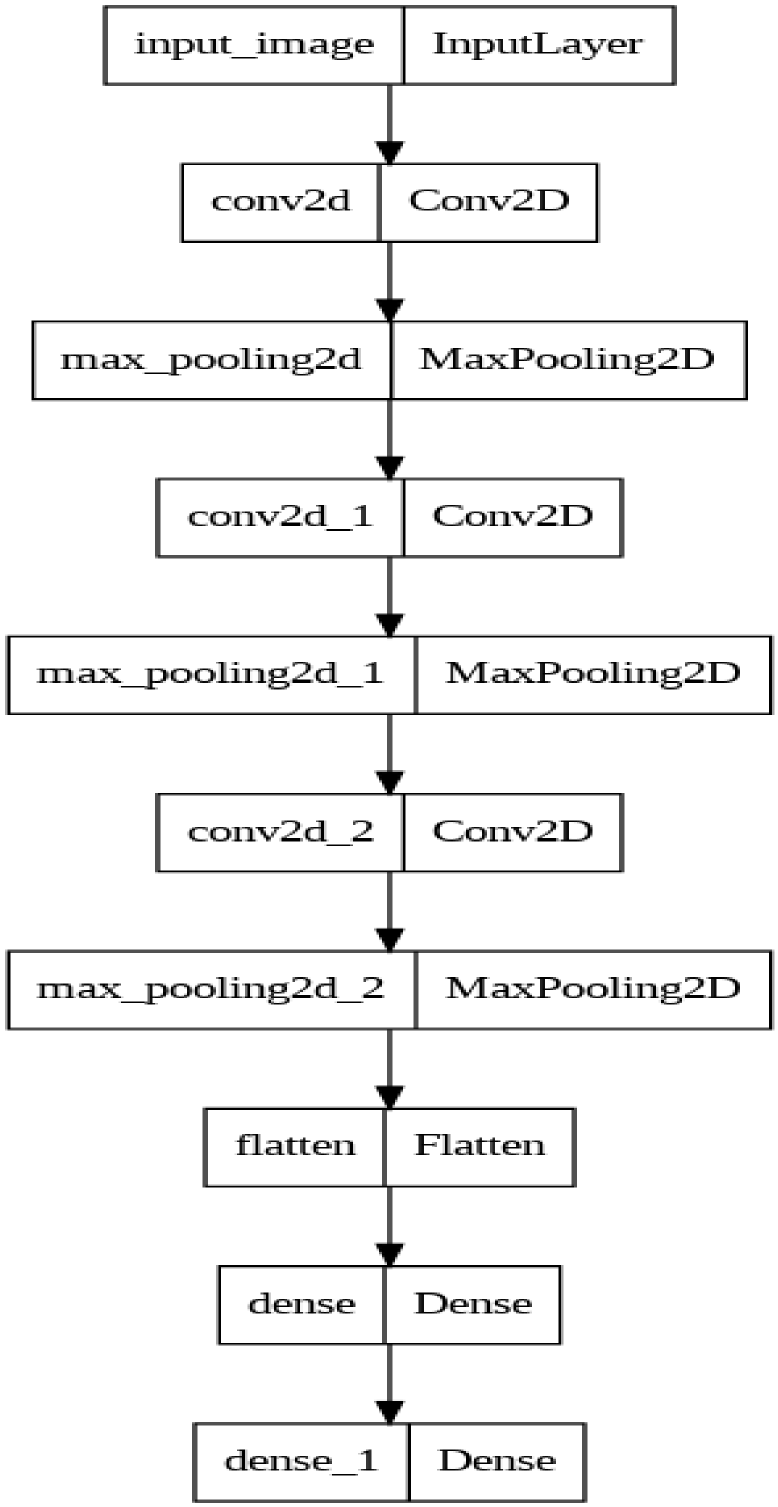

Figure 2: Base_model architecture

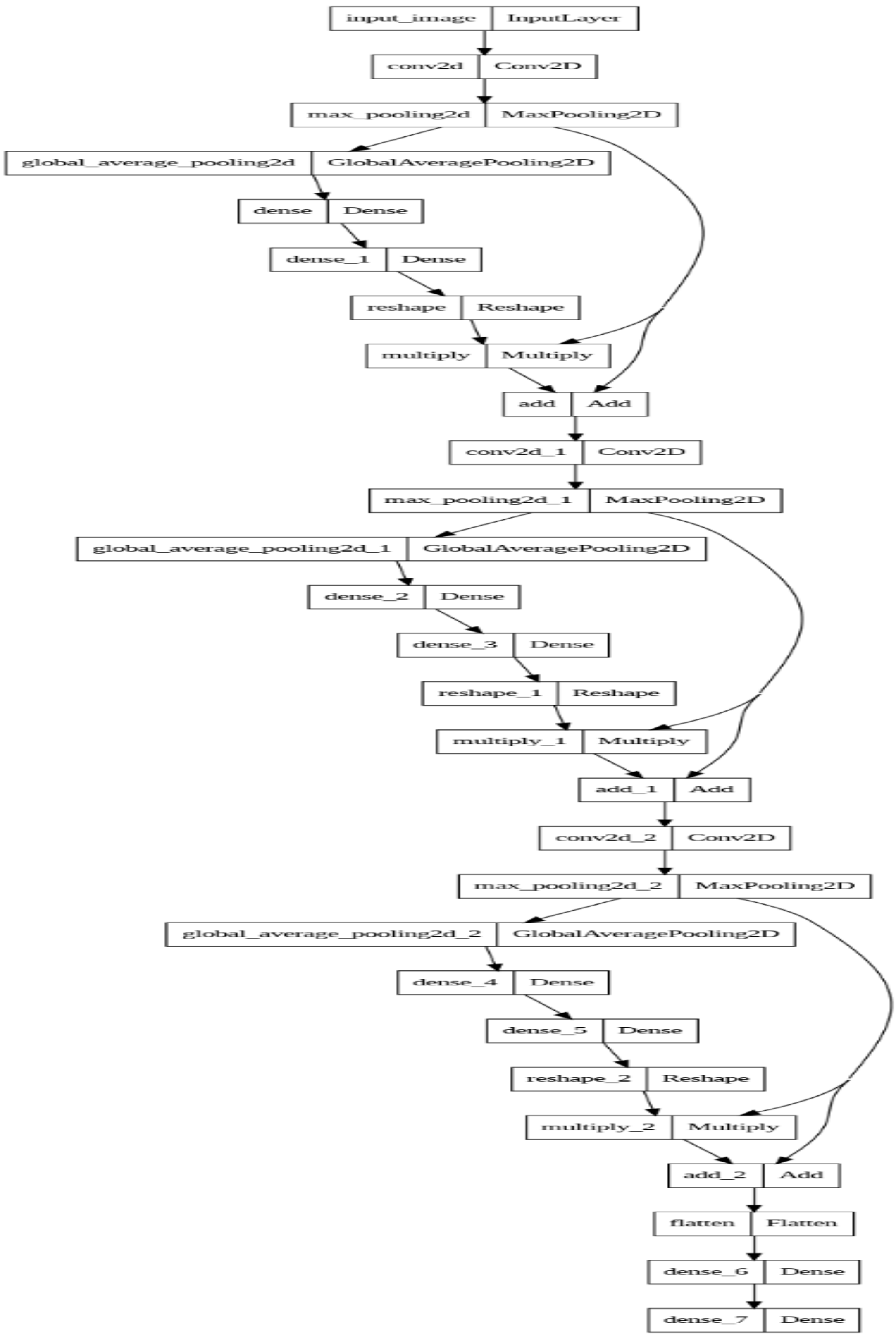

Figure 3: Model_v1 architecture
(Note: Figure is not visible clearly because image size does not fit on the page but in the project folder high-quality image is available)

- The training progress in terms of MAE and Val loss for both the models are shown in Figure 4 and Figure 5.

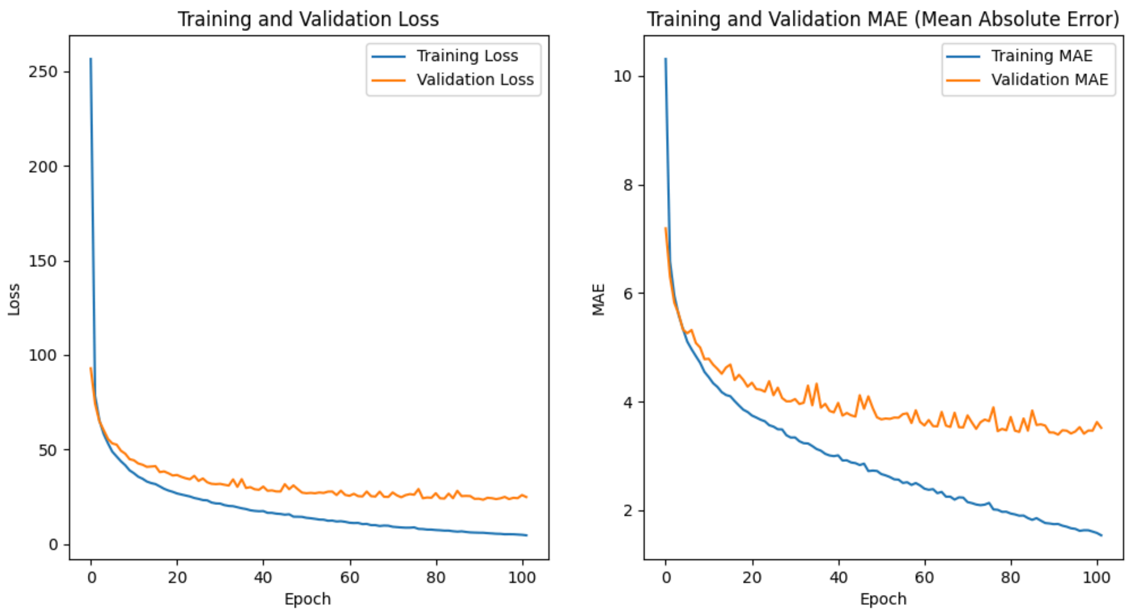

Figure 4: Base_model Training, validation loss and MAE (EarlyStop at 102 Epoch)

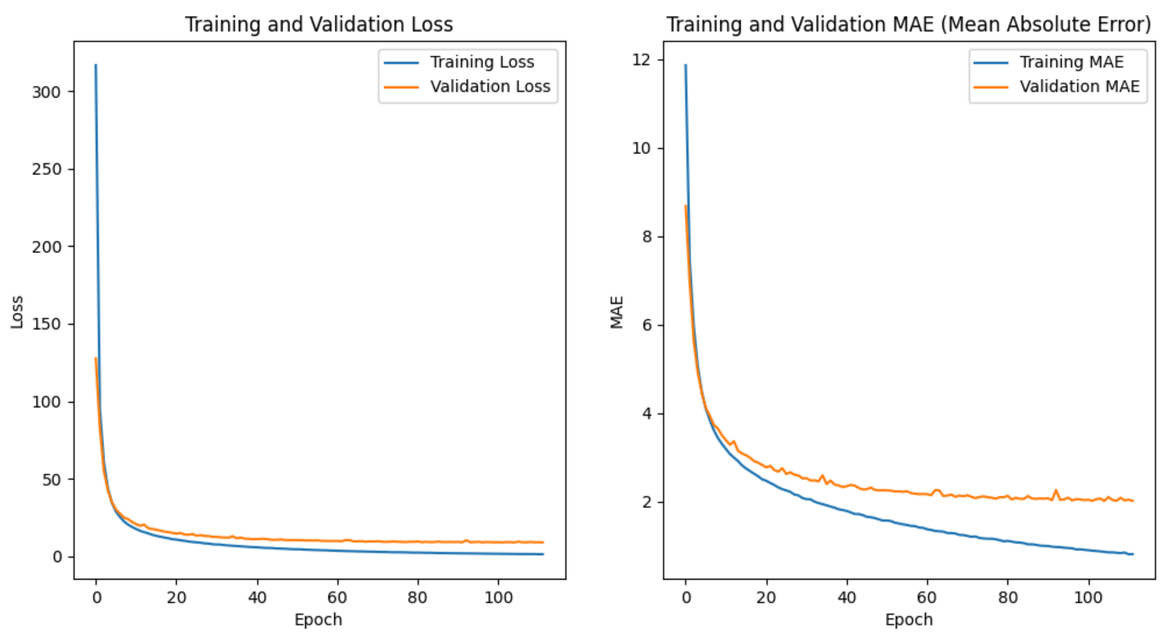

Figure 5: Mode_v1l Training, validation loss and MAE (EarlyStop at 112 Epoch)

- The sample predictions for the base_model are shown in Figure 6 and Figure 7 and the Model_v1 are shown in the Figure 8 and Figure 9.

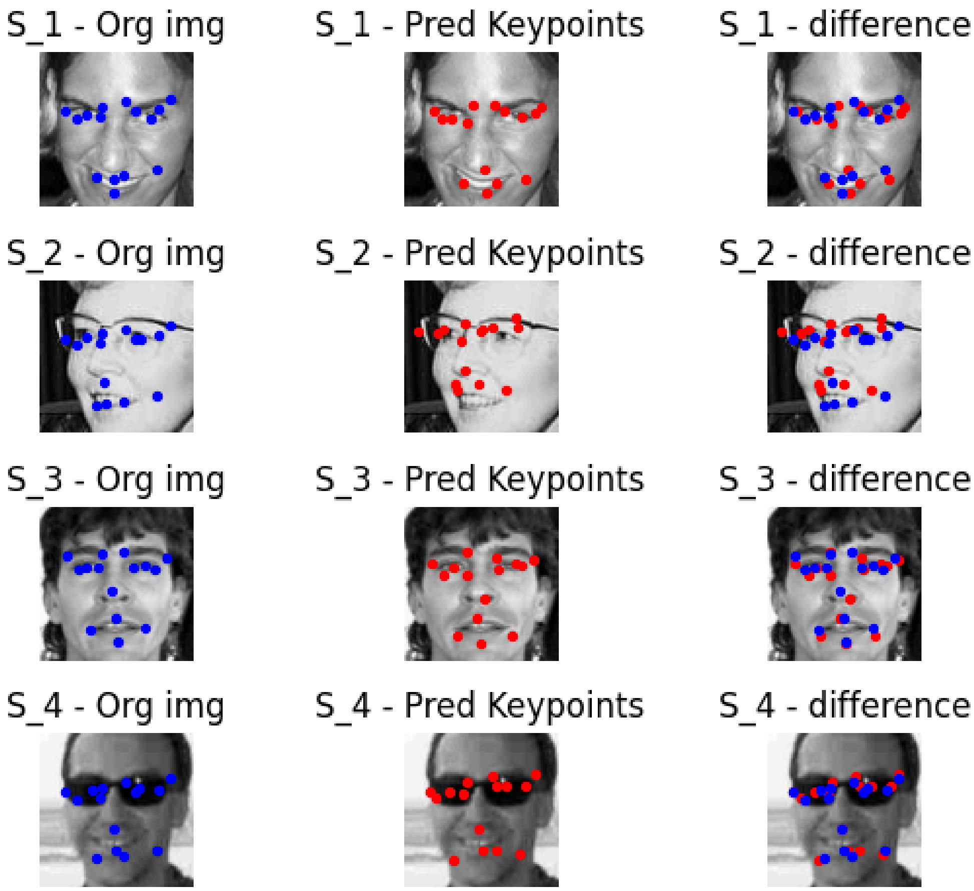

Figure 6: Base_model visualization (Column 1: Ground truth, Column 2: Predictions, and Column 3: Difference in the predictions)

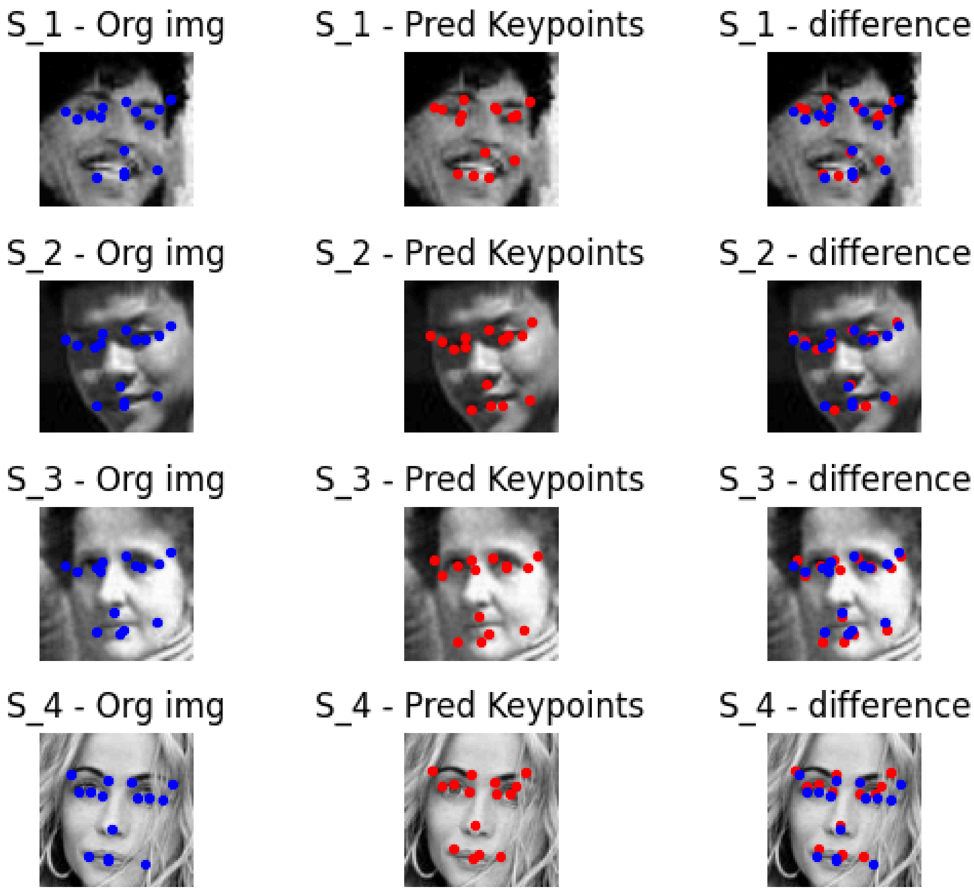

Figure 7: Base_model visualization (Column 1: Ground truth, Column 2: Predictions, and Column 3: Difference in the predictions)

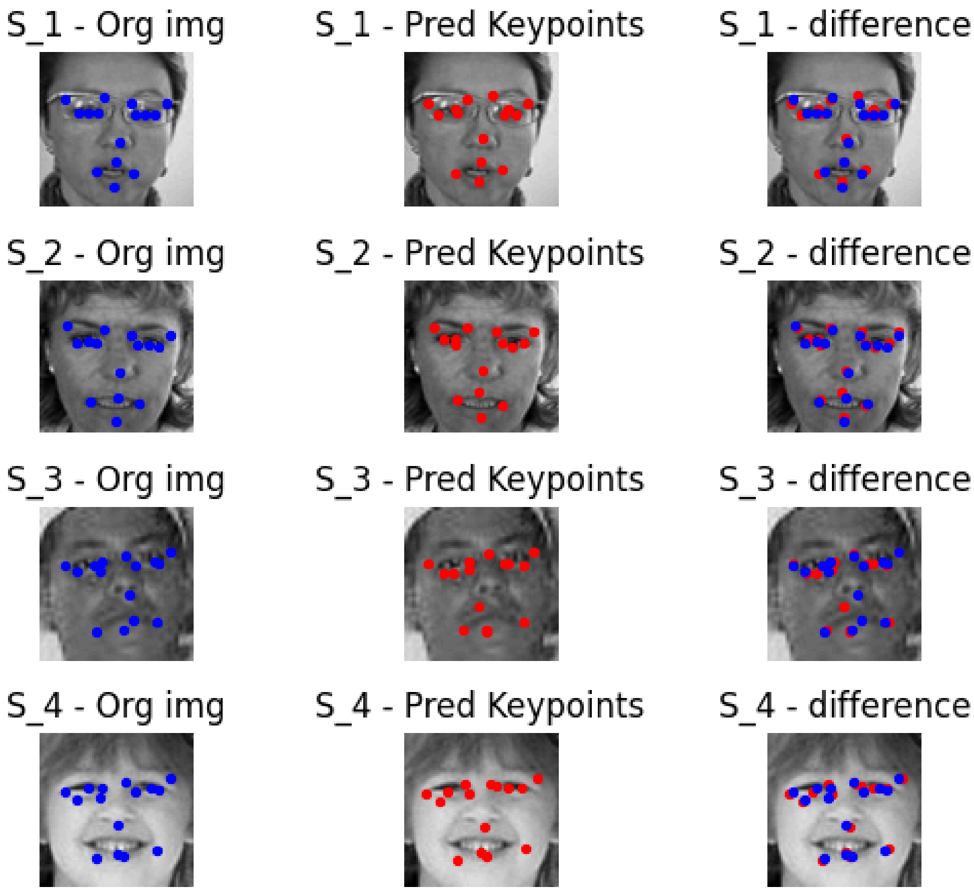

Figure 8: Model_v1 visualization (Column 1: Ground truth, Column 2: Predictions, and Column 3: Difference in the predictions)

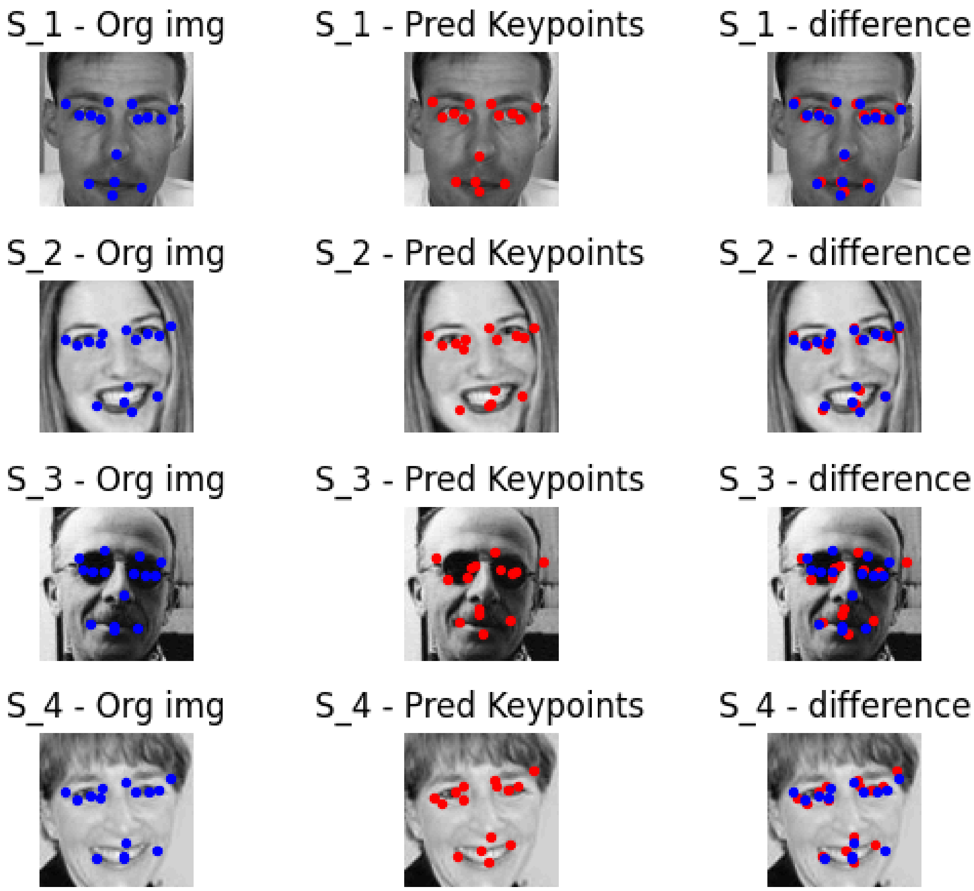

Figure 9: Model_v1 visualization (Column 1: Ground truth, Column 2: Predictions, and Column 3: Difference in the predictions)

- The visualization result indicates that the difference in the ground truth keypoints and the predicted keypoints is less in model_v1 compared the base_model.

Conclusion:

- From the result analysis and visual representation it is clear that the keypoint detection using the simple baseline approach is not outperforming.
- The introduction of the SENet in the network uplifts the keypoint detection performance but introduces complexity in the architecture design and increases the parameters.
- The model_v1 leverages the advantages of the channel-wise attention module and enhances the network learning and generalization ability of the model.
- The experimentation result analysis and the visualization results indicate that the model_v1 is outperforming the base_model.