Submitting:
Omri Arie
Liel Layney

Assignment 2 report Deep Learning

Siamese Network

Question 2 - Introduction

We decided to use 20% of the train data to validation.

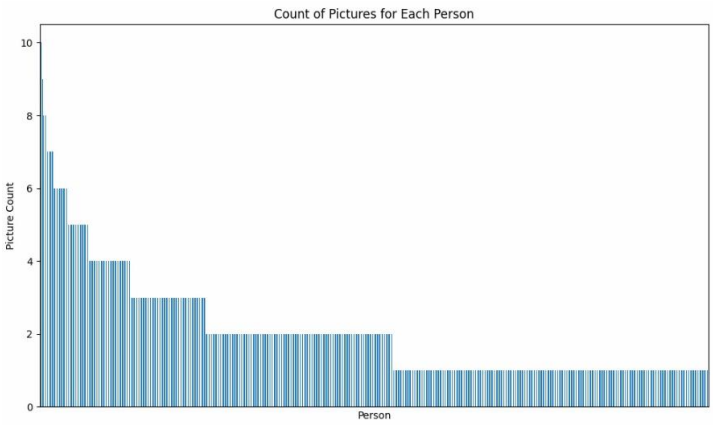
There are in the data about 5749 different people and about 4400 images in total.

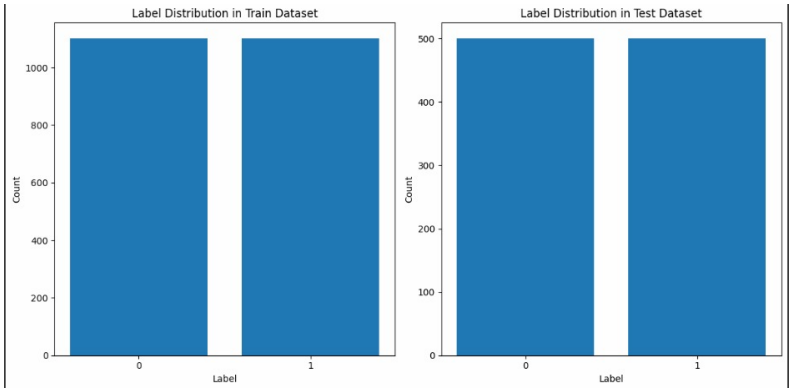
The images are in grey scale - i.e. one layer.

number of rows in train :2200

number of rows in test :1000

Each row represents two images that enter to the network for review. Sometimes they are images of the same person and sometimes of different people.





Submitting:
Omri Arie
Liel Layney

Implementation:

Base model from paper :

1. Experimental Setup

- **Batch Size**: 128 why? like in the paper .
- **Learning Rate**: 0.0001 why? The lower bound of the learning rate in the paper.
- **Validation Split**: 20% of the training data was used for validation. Why? Common separation.
- **Training Epochs**: 20 why? Minimal number to get to know the data
- **Stopping Criteria**: The model trained for a fixed number of 20 epochs. Why? We want the model to run all the 20 epochs like we plan.
- **Data Preprocessing**:
    - Images were resized to 105X105 pixels.
    - Images were normalized and converted to tensors.
- **Loss Function**: Binary Cross-Entropy Loss (BCELoss). why? like in the paper
- **Optimizer**: Adam optimizer with default parameters . why? Empirically the best optimizer
- **Run Time Length**: 5 min

2. Model Architecture

The Siamese network consists of the following components:

- **Convolutional Layers**:
    - **Conv1**: 64 filters, 10×10 kernel, stride 1, followed by ReLU activation and 2×2 max-pooling.
    - **Conv2**: 128 filters, 7×7 kernel, stride 1, followed by ReLU activation and 2×2 max-pooling.
    - **Conv3**: 128 filters, 4×4, stride 1, followed by ReLU activation and 2×2 max-pooling.
    - **Conv4**: 256 filters, 4×4 kernel, stride 1, followed by ReLU activation.
- **Fully Connected Layers**:
    - **FC1**: Input size 9216, output size 4096, followed by Sigmoid activation.
    - **Output Layer**: Input size 4096, output size 1, followed by Sigmoid activation.
- **Forward Pass**:
    - Absolute differences between image embeddings were passed through the output layer to produce similarity scores.

Submitting:
Omri Arie
Liel Layney

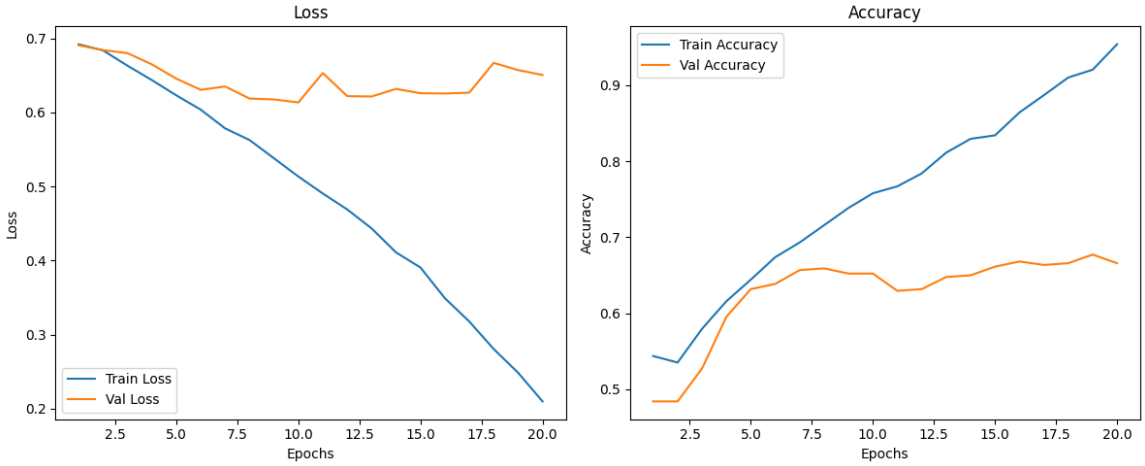
3. Reasoning Behind Design Choices

- **Architecture**:
  - The architecture was inspired by the referenced Siamese network paper, as been asked in the assignment.
- **Optimizer**:
  - Adam optimizer was selected for its adaptive learning rate capabilities and general robustness in training deep networks.
- **Learning Rate**:
  - A low learning rate (0.0001) was chosen to ensure stable convergence without overshooting.
- **Data Preprocessing**:
  - Resizing to 105×105 ensured compatibility with the original paper's architecture.

4. Performance Analysis

- **Training and Validation Metrics**:
  - The training loss consistently decreased, starting at 0.6922 in Epoch 1 and reaching 0.2098 in epoch 20.
  - Validation loss decreased initially but plateaued around epoch 15, indicating potential convergence or slight overfitting.
  - Validation accuracy improved from 48.41% to 66.82% by epoch 16 but fluctuated slightly thereafter.
- **Test Metrics**:
  - **Test Loss**: 0.7152
  - **Test Accuracy**: 66.00%
- **Observations**:
  - The model's performance on the test set (66% accuracy) suggests that it learned meaningful patterns but still has room for improvement.
  - Overfitting might have occurred towards the end of training, as the validation loss did not significantly improve after Epoch 16.

Submitting:
Omri Arie
Liel Layney



Improved model 1:

1.Experimental Setup

- **Batch Size**: 128 why? like in the paper .
- **Learning Rate**: 0.0001 why? The lower bound of the learning rate in the paper.
- **Validation Split**: 20% of the training data was used for validation. Why? Common separation.
- **Training Epochs**: 20 why? Minimal number to get to know the data
- **Stopping Criteria**: The model trained for a fixed number of 20 epochs. Why? We want the model to run all the 20 epochs like we plan.
- **Data Preprocessing**:
  - Random horizontal flips with a probability of 50%.
  - Random rotations up to ±15°.
  - Random adjustments to brightness (±20%) and contrast (±20%).
  - Images were resized to 105×105 pixels.
  - Images were normalized and converted to tensors.
- **Loss Function**: Binary Cross-Entropy Loss (BCELoss). why? like in the paper
- **Optimizer**: Adam optimizer with default parameters . why? Empirically the best optimizer
- **Run Time Length**: 7 min

2.Model Architecture

The Siamese network consists of the following components:

Submitting:
Omri Arie
Liel Layney

- **Convolutional Layers**:
  - **Conv1**: 64 filters, 10×10 kernel, stride 1, followed by BatchNorm, ReLU, and 2×2 max-pooling.
  - **Conv2**: 128 filters, 7×7 kernel, stride 1, followed by BatchNorm, ReLU, and 2×2 max-pooling.
  - **Conv3**: 128 filters, 4×4 kernel, stride 1, followed by BatchNorm, ReLU, and 2×2 max-pooling.
  - **Conv4**: 256 filters, 4×4 kernel, stride 1, followed by BatchNorm and ReLU.
- **Fully Connected Layers**:
  - **FC1**: Input size 9216, output size 4096, followed by ReLU.
  - **Output Layer**: Input size 4096, output size 1, followed by Sigmoid.
- **Forward Pass**:
  - Absolute differences between image embeddings were passed through the output layer to produce similarity scores.

3.Reasoning Behind Design Choices

- **Additions and Improvements**:
  1. **Batch Normalization**:
     - Added after each convolutional layer to normalize activations, stabilize training, and accelerate convergence.
  2. **Data Augmentation**:
     - Added random flips, rotations, and brightness/contrast adjustments to increase data diversity and improve generalization.
- **Comparison to Experiment 1**:
  - The key difference is the addition of batch normalization, which helped stabilize and speed up training.
  - Data augmentation techniques were introduced to mitigate overfitting and allow the model to learn from a more diverse dataset.
- **Justifications**:
  - Batch normalization and ReLU are well-known techniques to improve model training efficiency and reduce vanishing gradient issues.
  - Data augmentation was added to address potential overfitting observed in Experiment 1, where validation loss plateaued after epoch 15.

4. Performance Analysis

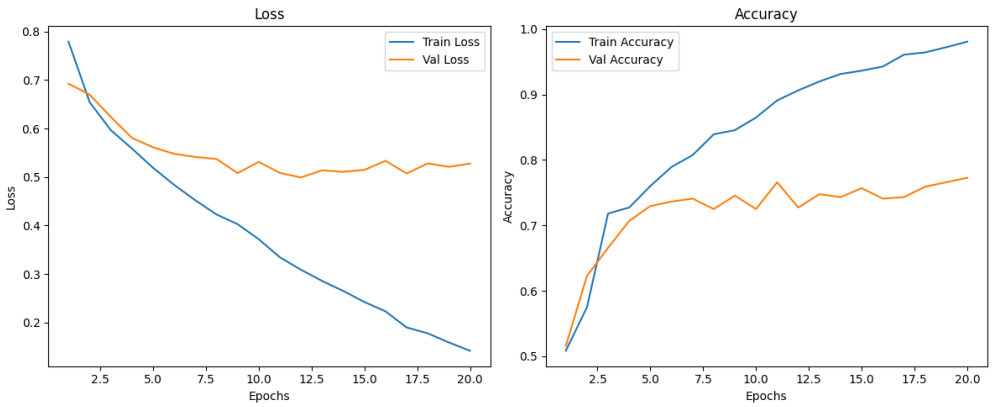
- **Training and Validation Metrics**:
  - The training loss consistently decreased, starting at 0.7791 in epoch 1 and reaching 0.1417 in epoch 20.
  - Validation loss showed a steady decline and performed better than Experiment 1, reaching 0.5073 at its lowest.

Submitting:
Omri Arie
Liel Layney

- Validation accuracy improved significantly, starting at 51.59% and reaching 77.27% by epoch 20.

- **Test Metrics**:
  - **Test Loss**: 0.5501
  - **Test Accuracy**: 75.00%

- **Observations**:
  - The model demonstrated improved performance compared to Experiment 1:
    - Validation accuracy increased from 66.82% to 77.27%.
    - Test accuracy increased from 66.00% to 75.00%.
  - The addition of batch normalization likely contributed to better generalization, while data augmentation improved the model's ability to handle variations in the test set.



Improved model 2:

1. Experimental Setup

- **Batch Size**: 128 why? like in the paper .
- **Learning Rate**: 0.0001, with StepLR scheduler reducing it by a factor of 0.1 every 5 epochs. why? To optimize the lower bound of the learning rate in the paper.
- **Validation Split**: 20% of the training data was used for validation. Why? Common separation.
- **Training Epochs**: 20 why? Minimal number to get to know the data
- **Stopping Criteria**: The model trained for a fixed number of 20 epochs. Why? We want the model to run all the 20 epochs like we plan.
- **Data Preprocessing**:
  - Random horizontal flips with a probability of 50%.
  - Random rotations up to ±15°.

Submitting:
Omri Arie
Liel Layney

  - Random adjustments to brightness (±20%) and contrast (±20%).
  - Images were resized to 105×105 pixels.
  - Images were normalized and converted to tensors.
- **Loss Function**: Binary Cross-Entropy Loss (BCELoss). why? like in the paper
- **Optimizer**: Adam optimizer with weight decay (1e−4) for L2 regularization. Why? for effective gradient-based optimization while mitigating overfitting
- **Learning Rate Scheduler**: StepLR reduced the learning rate by 90% every 5 epochs. Why? to dynamically reduce the learning rate for improved convergence during training.
- **Run Time Length**: 9 min

2.Model Architecture

The Siamese network consists of:

- **Convolutional Layers**:
  - Four convolutional blocks, each with BatchNorm, ReLU activation, and max-pooling:
    - Conv1: 64 filters, 10×10 kernel.
    - Conv2: 128 filters, 7×7 kernel.
    - Conv3: 128 filters, 4×44 kernel.
    - Conv4: 256 filters, 4×4 kernel.
- **Fully Connected Layers**:
  - FC1: Input size 9216, output size 4096, followed by ReLU and Dropout (p=0.5).
  - Output Layer: Input size 4096, output size 1, followed by Sigmoid.
- **Forward Pass**:
  - Absolute differences between image embeddings were passed through the output layer to produce similarity scores.

3. Changes from the last experiment and Reasoning Behind Design Choices

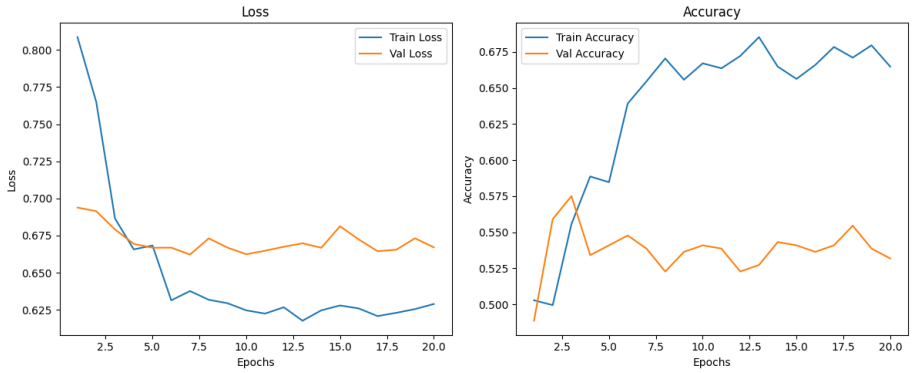
- **Additions and Improvements**:
  1. **Dropout in FC1**:
     - Added a 50% dropout rate to reduce overfitting by preventing co-adaptation of features.
  2. **Learning Rate Scheduler**:
     - Implemented StepLR to dynamically decrease the learning rate for better convergence.
  3. **L2 Regularization**:
     - Introduced weight decay in the optimizer to penalize large weights and reduce overfitting.

Submitting:
Omri Arie
Liel Layney

- **Comparison to Experiment 2**:
  - The addition of dropout aimed to address potential overfitting observed in the validation set of Experiment 2.
  - A learning rate scheduler was included to allow the model to fine-tune weights during later epochs.
  - Weight decay added explicit regularization, which was absent in Experiment 2.
- **Justifications**:
  - Dropout and L2 regularization are common techniques to improve generalization, especially in fully connected layers.
  - The learning rate scheduler ensures stable convergence by reducing the learning rate as training progresses.

4. Performance Analysis

- **Training and Validation Metrics**:
  - Training loss started at 0.8086 and plateaued around 0.6208 by epoch 17.
  - Validation loss did not decrease significantly, fluctuating around 0.6645 toward the end of training.
  - Validation accuracy peaked early at 57.5% but declined to 53.18% in the later epochs, indicating potential underfitting.
- **Test Metrics**:
  - **Test Loss**: 0.6596
  - **Test Accuracy**: 52.90%
- **Observations**:
  - Despite the changes, the model underperformed compared to Experiment 2:
    - Validation and test accuracy declined significantly.
    - The learning rate scheduler may have reduced the learning rate too aggressively, causing stagnation in learning after the initial epochs.
  - The addition of dropout and weight decay might have been overly penalizing for the model, leading to underfitting.



Submitting:
Omri Arie
Liel Layney

Improved model 3

1. Experimental Setup

- **Batch Size**: 128 why? like in the paper .
- **Learning Rate**: 0.0001, with StepLR scheduler reducing it by a factor of 0.1 every 5 epochs. why? To optimize the lower bound of the learning rate in the paper.
- **Validation Split**: 20% of the training data was used for validation. Why? Common separation.
- **Training Epochs**: 20 why? Minimal number to get to know the data
- **Stopping Criteria**: Early stopping was triggered after 11 epochs due to no improvement in validation loss for 5 consecutive epochs. Why? to prevent overfitting and save training time by halting when validation loss showed no improvement for 5 consecutive epochs.
- **Data Preprocessing**:
    - Random horizontal flips with a probability of 50%.
    - Random rotations up to ±15°.
    - Random adjustments to brightness (±20%) and contrast (±20%).
    - Images were resized to 105×105 pixels.
    - Images were normalized and converted to tensors.
- **Loss Function**: Mean Squared Error Loss (MSELoss) instead of Binary Cross-Entropy Loss (BCELoss). Why ? Trial and error
- **Optimizer**: Adam optimizer with default parameters . why? Empirically the best optimizer
- **Run Time Length**: 5 min

2. Model Architecture

The Siamese network consists of:

- **Convolutional Layers**:
    - Four convolutional blocks, each with BatchNorm, ReLU activation, and max-pooling:
        - Conv1: 64 filters, 10×10 kernel.
        - Conv2: 128 filters, 7×7 kernel.
        - Conv3: 128 filters, 4×4 kernel.
        - Conv4: 256 filters, 4×4 kernel.
- **Fully Connected Layers**:
    - FC1: Input size 9216, output size 4096, followed by ReLU.
    - Output Layer: Input size 4096, output size 1, followed by Sigmoid.
- **Forward Pass**:

Submitting:
Omri Arie
Liel Layney

- Absolute differences between image embeddings were passed through the output layer to produce similarity scores.

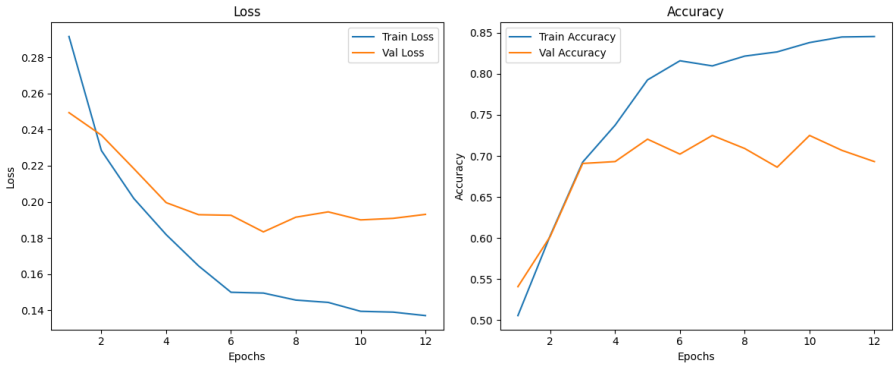
3. Reasoning Behind Design Choices

- **Additions and Improvements**:
  1. **Early Stopping**:
     - Implemented to halt training when validation loss did not improve for 5 consecutive epochs, preventing overfitting and saving computational resources.
  2. **Loss Function**:
     - Switched to MSELoss for better gradient behavior and smoother updates in training.
  3. **Learning Rate Scheduler**:
     - Retained StepLR to dynamically reduce the learning rate for gradual convergence.
- **Comparison to Experiment 3**:
  - Early stopping was introduced to prevent overfitting, which was observed in Experiment 3 as validation loss plateaued.
  - MSELoss replaced BCELoss, aiming for a smoother optimization process and potentially improved convergence.
  - The removal of Dropout in the fully connected layers ensured better representation learning by avoiding excessive regularization.

4. Performance Analysis

- **Training and Validation Metrics**:
  - Training loss steadily decreased from 0.2915 to 0.1390 over 11 epochs.
  - Validation loss consistently improved until epoch 7, reaching its best value of 0.1834, before plateauing.
  - Validation accuracy peaked at 72.50% at epoch 10, showing a significant improvement over Experiment 3.
- **Test Metrics**:
  - **Test Loss**: 0.5455
  - **Test Accuracy**: 72.60%
- **Observations**:
  - Early stopping effectively prevented overfitting, leading to improved generalization.
  - Test accuracy improved significantly compared to Experiment 3 (52.90% to 72.60%), indicating that the model benefitted from the changes.
  - Switching to MSELoss contributed to a smoother and more stable training process.

Submitting:
Omri Arie
Liel Layney



Summary of Experiments

The experiments were conducted with progressively refined architectures and training strategies to improve the Siamese network's performance on the given dataset. Below is a summary of each experiment:

1. **Experiment 1**:
   - **Setup**: Base architecture from the reference paper.
   - **Key Results**:
     - Test Loss: 0.7152
     - Test Accuracy: 66.00%
   - **Observations**: The model exhibited steady learning, but validation accuracy plateaued, indicating room for optimization in generalization and training dynamics.
2. **Experiment 2**:
   - **Changes**: Added batch normalization, data augmentation, and replaced Sigmoid activation in fully connected layers with ReLU.
   - **Key Results**:
     - Test Loss: 0.5501
     - Test Accuracy: 75.00%
   - **Observations**: Improved test accuracy and reduced test loss demonstrated the effectiveness of normalization and augmentation. Validation metrics showed consistent improvement.
3. **Experiment 3**:
   - **Changes**: Introduced L2 regularization, dropout in fully connected layers, and a learning rate scheduler.
   - **Key Results**:
     - Test Loss: 0.6596
     - Test Accuracy: 52.90%
   - **Observations**: Over-regularization led to underfitting, as reflected in lower validation and test performance. Learning rate scheduling may have decreased the rate too aggressively.

Submitting:
Omri Arie
Liel Layney

4. **Experiment 4**:
    - **Changes**: Switched to MSELoss, introduced early stopping, and fine-tuned learning rate adjustments.
    - **Key Results**:
        - Test Loss: 0.5455
        - Test Accuracy: 72.60%
    - **Observations**: Early stopping effectively balanced training duration and generalization, while MSELoss stabilized training. The model achieved its best validation performance early, showcasing the utility of adaptive stopping criteria.

Conclusions

1. **Impact of Batch Normalization and Augmentation**:
    - These additions (Experiment 2) significantly improved the model's generalization and learning efficiency, resulting in the best test accuracy (75.00%) among all experiments.
2. **Regularization Challenges**:
    - Experiment 3 highlighted the risk of excessive regularization through dropout and L2 regularization, which caused underfitting and degraded performance.
3. **Utility of Early Stopping**:
    - Experiment 4 demonstrated that early stopping prevents overfitting and saves computational resources, with comparable test accuracy to Experiment 2.
4. **Loss Function Insights**:
    - MSELoss in Experiment 4 provided smoother gradients, aiding training stability compared to BCELoss.

Overall Summary

The experiments revealed that:

- **Batch normalization, data augmentation, and ReLU activations** are critical for improving model performance.
- **Over-regularization** can harm learning, emphasizing the need for balance in applying techniques like dropout and L2 regularization.
- **Early stopping** and an appropriately chosen loss function stabilize training and improve generalization.

The final architecture (Experiment 4) delivered robust and balanced performance with test accuracy of 72.60%, closely aligning with the highest accuracy achieved (75.00%)

Submitting:
Omri Arie
Liel Layney
while offering better training efficiency. These findings underscore the importance of iterative refinement and empirical evaluation in model development.

Grid Search and Results

To optimize the hyperparameters of the Siamese neural network model, a grid search was performed over a comprehensive parameter space. The grid search included variations in the number of epochs, batch size, learning rate, and early stopping patience:

- **Parameters Explored**:
    - Epochs: [10, 20, 30]
    - Batch Size: [64, 128, 256]
    - Learning Rate: [0.001, 0.0001, 0.00001]
    - Early Stopping Patience: [3, 5, 7]
- **Dataset Splitting and Augmentation**:
  The dataset was split into training (80%) and validation (20%) subsets. Data augmentation techniques such as random horizontal flips, rotations, and brightness/contrast adjustments were applied to improve generalization.
- **Evaluation Criteria**:
  Models were evaluated based on validation loss, and the best-performing model was selected using early stopping.

Optimal Configuration and Validation Results

The optimal configuration was:

- **Epochs**: 20
- **Batch Size**: 64
- **Learning Rate**: 0.0001
- **Early Stopping Patience**: 3

This setup achieved the lowest validation loss of **0.1671**, indicating superior performance on the validation set. The runtime took 4 hours.

Submitting:
Omri Arie
Liel Layney

Test Results

The best model was further evaluated on a separate test set, yielding the following metrics:

- **Test Loss**: 0.5332
- **Test Accuracy**: 72.80%

These results demonstrate the model's effectiveness in generalizing to unseen data, making it a reliable baseline for future improvements.

Submitting:
Omri Arie
Liel Layney

Examples of the model classification

Correct Classification:

The model successfully identified these two images as belonging to the same individual. Key reasons for the correct classification include:

1. **Facial Similarity**: Both images share distinct facial features, such as a mustache and similar face shapes, which the model likely used as strong indicators for matching.
2. **Angle Consistency**: The images are taken from similar angles, reducing the complexity of the spatial features that the model needs to align and compare.
3. **Clear Visual Features**: Both images are well-lit and have clear details of the person's face, minimizing noise and aiding the convolutional layers in extracting meaningful features.

Correct Classification:

The model successfully identified these two images as belonging to the same individual. Factors contributing to the correct classification include:

1. **Distinct Facial Features**: Both images display key characteristics, such as a similar mustache style and face shape, which likely served as strong cues for the model to match the pair.
2. **Facial Orientation**: Although one image shows the individual with closed eyes, the overall facial orientation remains consistent, making it easier for the model to align and compare features.

Submitting:
Omri Arie
Liel Layney

Correct Classification:

The model accurately identified these two images as belonging to the same individual. Key factors for the successful classification include:

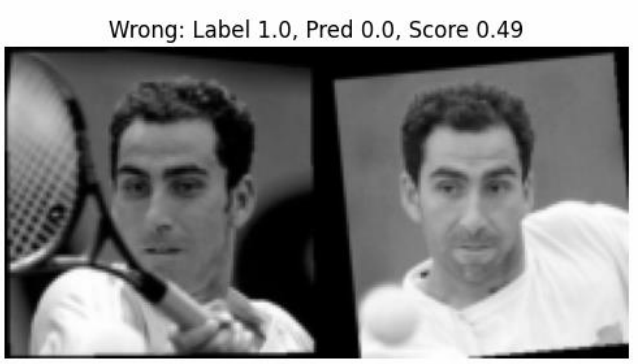
1. **Prominent Facial Features**: The individual has distinctive characteristics, such as a similar hairline, eye shape, and facial structure, which provided strong cues for the model to establish similarity.
2. **Expression Neutrality**: Both images display neutral facial expressions, reducing variability in the facial features for comparison and making it easier for the model to focus on consistent traits.
3. **Consistent Head Positioning**: The head orientation and alignment are quite similar across the two images, aiding the model in matching spatially aligned features effectively.

Incorrect Classification:

The model misclassified this pair as belonging to different individuals (Label 1.0, Pred 0.0). The reasons for this misclassification are likely to include:

1. **Presence of Additional Faces**: One of the images contains multiple faces in the background, which could have introduced noise and confused the model during feature extraction. The secondary face may have diverted attention from the primary individual.
2. **Significant Angle Variation**: The two images show the individual from markedly different angles—one is a frontal view while the other is a profile view. This variation can make it challenging for the model to align and compare features effectively.
3. **Obstructed Facial Features**: In the second image, parts of the face are partially obscured or less prominent due to the angle and the interaction with another individual, reducing the visibility of critical features like the jawline and mouth.

Submitting:
Omri Arie
Liel Layney



Incorrect Classification:

The model incorrectly classified this pair as belonging to different individuals (Label 1.0, Pred 0.0). However, the score (0.49) indicates the model was close to making the correct prediction. Key reasons for the misclassification include:

1. **Presence of Distracting Objects**: The tennis racket in one image introduces additional patterns that may have distracted the model during feature extraction, diverting its focus from the individual's facial features.
2. **Angle and Pose Variability**: The two images present the individual at significantly different angles, with one showing a more side-profile view while the other is closer to a frontal view. This variability can reduce the model's ability to align and compare facial features effectively.

Incorrect Classification:

The model incorrectly classified this pair as belonging to different individuals (Label 1.0, Pred 0.0). The low similarity score (0.33) highlights the challenges the model faced in recognizing the individual. Key reasons for the misclassification include:

1. **Presence of a Head Covering**: In one image, the individual is wearing a head covering, which obscures parts of the face and hairline that the model typically relies on for feature extraction and comparison. This likely made it difficult for the model to detect matching patterns between the two images.
2. **Facial Context Variability**: The head covering in one image shifts the visual context of the face, making it appear significantly different from the unobstructed image. The model might have interpreted the head covering as part of the individual's overall features, leading to confusion.
3. **Background Distraction**: One image includes another person's face in the background, which could have introduced noise during feature extraction. The presence of additional facial patterns might have caused the model to misattribute features to the incorrect individual.