This solution employs a Siamese Neural Network to perform face recognition.

The key components and processes are outlined below:

**Data Collection and Preparation**:

* Data consists of face images from 'extracted\_faces' directory.
* Images are resized to 128x128 pixels and normalized to [0, 1] range.
* Esch photo has its own label, which depends on the folder in which it is located

**Model Architecture**:

* A Siamese Neural Network is employed, consisting of shared convolutional layers followed by fully connected layers.

*Architecture:*

- Input layer with the specified input shape.

- Convolutional layer with 32 filters and a 5x5 kernel, ReLU activation.

- MaxPooling layer with 64 filters and 2x2 pooling

- Convolutional layer with 64 filters and a 4x4 kernel, ReLU activation.

- MaxPooling layer with 32 filters and 2x2 pooling

- Convolutional layer with 64 filters and a 4x4 kernel, ReLU activation.

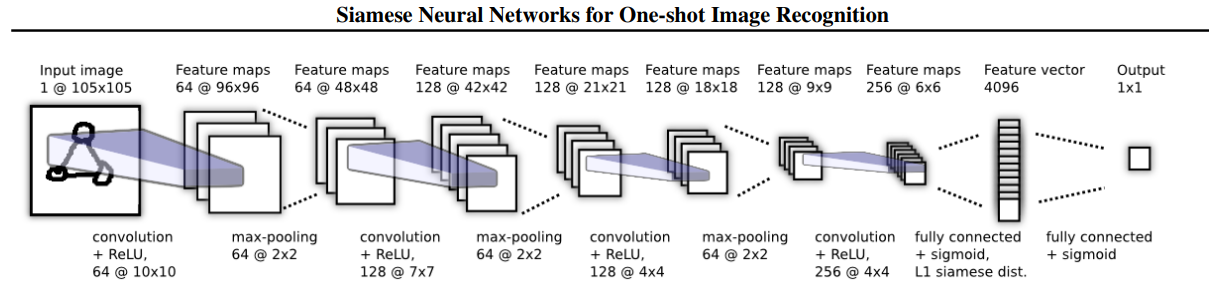
- MaxPooling layer with 32 filters and 2x2 pooling

- Convolutional layer with 128 filters and a 4x4 kernel, ReLU activation.

- Flatten layer to transform the 2D feature maps into a 1D vector.

- Fully connected dense layer with 128 units and sigmoid activation.

The architecture similar to “Siamese Neural Networks for One-shot Image Recognition” by Gregory Koch, Richard Zemel, Ruslan Salakhutdinov [https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf]:



* The network learns to generate embeddings (feature vectors) for faces.

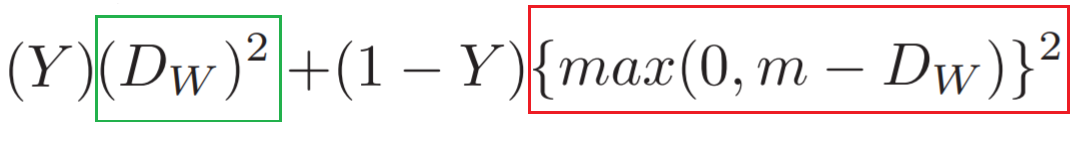
**Pair Generation**:

* Positive and negative pairs of face images are generated for training
* Positive pairs contain images of the same person, and negative pairs contain images of different people
* Randomization is used to create diverse pairs

**Training Process**:

* Training occurs over several epochs with a batch size of 64
* A contrastive loss function is used to encourage embeddings of positive pairs to be close and those of negative pairs to be distant

**Сontrastive loss function:**



* Adam optimizer is employed for weight updates

**Note:** also wanted to use triplets and triplet loss in my model, but I couldn't fully grasp the concept and implementation of this task

**Data Augmentation**:

* Data augmentation is applied to each batch during training to improve model generalization
* Augmentations include random rotation, horizontal and vertical shifts, and horizontal flips

**Inference and Face Recognition**:

* Faces are detected in images using the MTCNN (Multi-task Cascaded Convolutional Networks) face detection model
* Detected faces are compared with known individuals based on their embeddings
* The recognized individual is labeled, and bounding boxes are drawn around their faces

**Attention:**

My SNN-model is poorly trained and it faces several challenges. As a result, the assigned task is not fully completed. I have made an effort to accomplish the task given constraints.