Assignment 1B –

Person Re-Identification & Multi-Task Learning

CAB420 – Machine Learning

Gregory Mandall - n10757163

Submitted: May 2024

\* Code adapted from CAB420 lectures and practical solutions

PERSON RE-IDENTIFICATION

1. **Pre-processing**

The Market-1501 dataset contains several issues including viewpoint variations, class imbalance, background clutter and obstructions, and potential lighting issues. This may lead to performance fluctuations due to variations in appearance and pose, disparate frequency of images per identity and the presence of complex backgrounds, making accurate identification increasingly challenging. Image quality is documented and explored further in the analysis section of this report. Figure 1 depicts the distribution of images across each class present in the dataset, visualising the class imbalance present in the dataset.

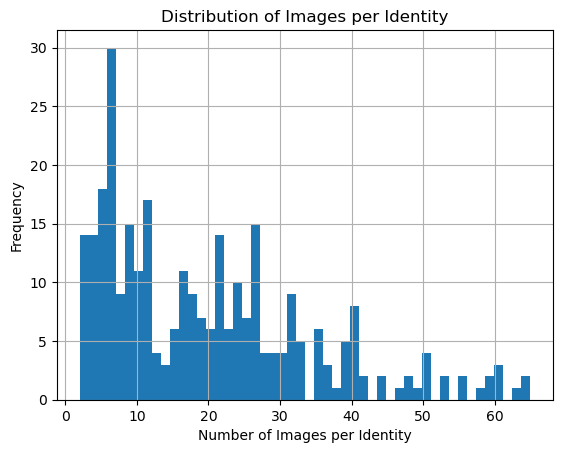


Figure 1: Distribution of Images per Identity to Visualise Class Imbalance

Data augmentation was implemented to address these issues and enhance the performance of the deep-learning and non-deep-learning methods. The data augmentation techniques applied included rotation up to 10 degrees, width and height shifts up to 10%, zoom up to 10%, horizontal flipping, and brightness modifications varying from 80% to 120% of the original brightness level. Implementing these methods introduced realistic variations that may potentially occur in real-world scenarios. Augmentation artificially expanded and increased the diversity of the training dataset without the need for additional data, attempting to promote increased generalisation and performance.

Images were resized to 64x32 to increase computational efficiency while maintaining adequate performance levels. Images were kept in colour to retain critical distinguishing features crucial for recognising individuals based on clothing or accessory colours, which are significant identifiers in person re-identification tasks.

1. **Model Details**

Principal Component Analysis (PCA) was selected over Linear Discriminant Analysis (LDA) for the non-deep-learning method due to the dataset’s structure. Several instances have a limited number of images representing the identity. This may limit the effectiveness of LDA, as LDA requires multiple samples per class to compute between-class and within-class variance effectively. Therefore, PCA’s approach of reducing dimensionality while retaining a significant portion of variance was deemed suitable for handling the variations in the Market-1501 dataset. A PCA subspace was generated using augmented, vectorised training data and projected onto the gallery and probe sets. The cumulative sum of explained variance was used to determine the optimal number of features to retain that explain 95% of the variance, leading to the retention of 107 components. This threshold was selected to maintain a balance between dimensionality reduction and information retention by disregarding less informative features. Visualising the cumulative variance, shown in Figure 2, validated that 107 components were sufficient to capture essential data characteristics, facilitating a comparable representation of the dataset without the computational overhead of handling higher-dimensional data.

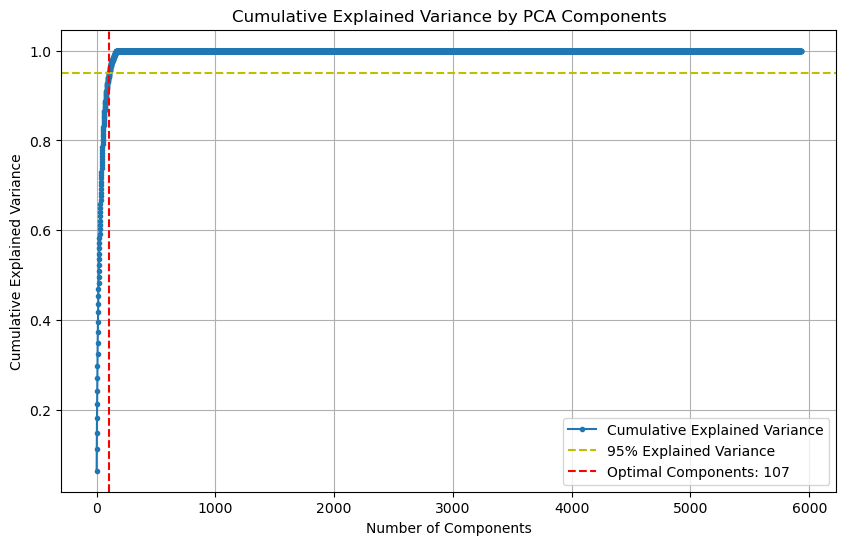


Figure 2: Cumulative Variance versus Number of Components to Visualise Optimal Feature Selection and Data Dimensionality.

The deep-learning method utilised a triplet loss network with a deep convolutional neural network (DCNN) backbone to optimise feature learning. This approach employed the strengths of DCNNs, which excel in processing pixel data from images. Triplet loss was implemented due to its increasing effectiveness in learning fine-grained differences between complex classes relative to contrastive loss. This is essential in a dataset with intra-class variations like Market-1501. The neural network utilises a tailored VGG-like architecture. This design was selected as dense layers lead to slightly superior performance and enhanced computational efficiency compared to a ResNet in scenarios with lower filter counts involved in model development. The network consists of three main convolutional blocks. The first block includes two Conv2D layers with 16 filters of 7x7 kernel size. The larger kernel size in the initial block was chosen to capture broad image features early in the network. Each convolutional layer is followed by batch normalisation to reduce computation time. The block ends with a MaxPooling2D layer to reduce feature map dimensionality. The size of the filters decreases to 5x5 and then to 3x3 while increasing the number of filters to 32 and then to 64 as the network progresses. This allows the network to capture increasingly detailed features necessary for person re-identification. The convolution blocks are flattened, and the vector is passed to a dense layer of 300 units.

Embeddings are normalised to ensure that the magnitude of each embedding vector is consistent, facilitating stable distance calculations between vectors. Normalisation is critical for the triplet loss function to effectively measure the relative distances necessary for distinguishing between disparate identities in the Market-1501 dataset. Swish and ReLU activation functions were compared during experimentation. Ultimately, Swish activation was selected for the final network due to its superior performance metrics. A complete model breakdown is shown in Table 1 of the Appendix. The triplet loss margin was set to 1.0 to define the distance to distinguish between dissimilar items in the feature space. Adam optimiser was utilised and a batch size of 350 was selected to ensure diverse class representation within each batch, optimising learning and enhancing the accuracy of gradient estimates during updates. The training extends over 40 epochs to balance between adequate convergence of the loss function and training time. Figure 3 showcases the training and validation loss curves, providing insight into the model’s learning dynamics. The plot indicates that stability and convergence is achieved at approximately 35 epochs.

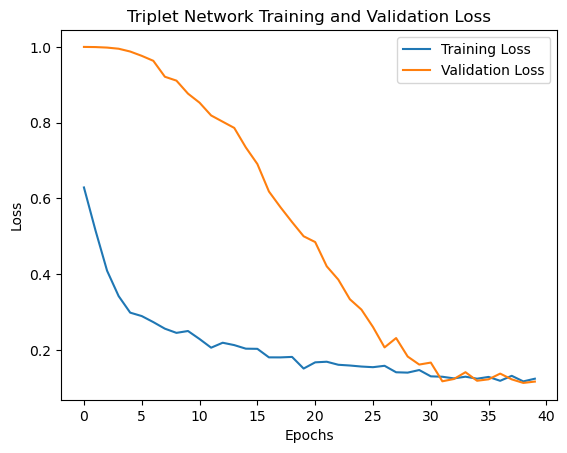


Figure 3: Loss versus Epochs to Determine Successful Convergence of the Triplet Network.

1. **Evaluation and Analysis**

The comparison between the PCA and Triplet Loss methods reveals significant disparities in performance across various scenarios. The Triplet Loss model demonstrates superior performance across all accuracy measures. This exhibits a clear advantage for the deep learning approach, potentially due to the method’s ability to handle increasingly complex image variations. The Top-1, Top-5, and Top-10 accuracies are shown in Table 1 below.

Table 1: Top-1, Top-5, and Top-10 Accuracy for the PCA and Triplet Loss Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Top-1 Accuracy** | **Top-5 Accuracy** | **Top-10 Accuracy** |
| *PCA* | 0.14 | 0.23 | 0.35 |
| *Triplet Loss* | 0.25 | 0.48 | 0.62 |

Instances illustrating the superior performance of the deep-learning method can be observed in a significant quantity of images present in the gallery and probe datasets. Images such as ID #1423 demonstrate the model’s proficiency in adjusting to lighting changes that alter clothing hues. Image ID #1259 showcases the model’s ability to maintain high accuracy measures despite significant distortions between the gallery and probe images. Figure 4 illustrates the robustness of the deep learning model, in which the graph delineates a consistent trend where deep learning outperforms PCA. The plot displays higher accuracy rates for individual IDs across a range of scenarios. Specific examples of instances where the triplet network outperforms PCA are shown in Figure 5 below. This suggests that the triplet loss network’s superiority is particularly evident in scenarios involving increasingly convoluted images, facial or image distortions, and minor lighting variations that seem to affect clothing colours. However, PCA outperforms the triplet network across several instances where images from the probe and gallery sets exhibit uniformity in lighting, minimal distortion, and maintain consistent image quality. This showcases the model’s simplicity and efficiency in processing less complex and highly uniform scenarios. Figure 6 depicts instances where PCA demonstrates superior performance under these conditions.

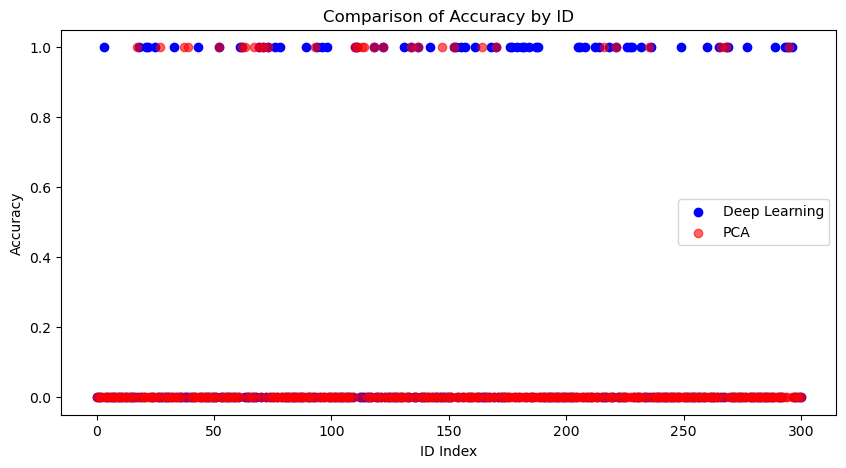


Figure 4: Comparison of Accuracy by ID to Visualise the Performance of Deep-Learning and Non-Deep-Learning Methods.



Figure 5: Instances where the Triplet Loss Network Significantly Outperforms PCA



Figure 6: Instances where PCA Outperforms the Triplet Loss Network.

The CMC curves for the Triplet Loss network and Principal Component Analysis reflect the disparities observed in the accuracy metrics and provide further insight into the comparative performance of the two methods. The curve for the Triplet Loss method exhibits a steeper rise, showcasing this method’s ability to differentiate between similar and dissimilar samples with increasing accuracy relative to PCA, a critical capability for pattern recognition tasks. Additionally, the shallower slope of the PCA’s curve may suggest that this method faces challenges in distinguishing between closely related classes. However, the plots suggest that each method converges to near 100% accuracy at approximately the same point. This convergence underscores the importance of both methodologies, highlighting that each may be strategically employed depending on specific use case requirements. Figure 7 visualises the CMC curves for the deep-learning and non-deep-learning methods.

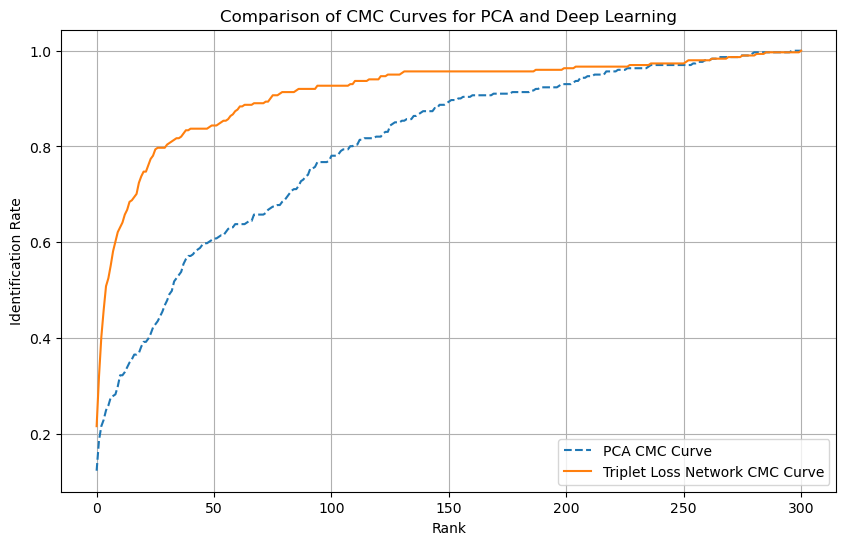


Figure 7: CMC Curve for PCA and the Triplet Loss Network to Provide a Graphical Representation of Each Method’s Performance.

PCA demonstrates significantly lower training and inference times relative to the Triplet Loss method. This demonstrates the computational efficiency of the method, making it increasingly suitable for use case scenarios where computational resources are limited, despite its lower overall accuracy. However, person re-identification tasks require high accuracy. This necessitates the consideration of deep learning methods despite longer training and inference times, due to their ability to handle complex image data and achieve higher accuracy compared to traditional machine learning methods. Table 2 displays the training and inference times for the Triplet Loss Network and Principal Component Analysis.

Table 2: Training Time and Inference Time for PCA and Triple Loss Methods.

|  |  |  |
| --- | --- | --- |
| **Method** | **Training Time, seconds (s)** | **Inference Time, seconds (s)** |
| *PCA* | 166.27 | 0.01 |
| *Triplet Loss* | 627.33 | 5.03 |

1. **Ethical Considerations**

Research into person re-identification is crucial in advancing facial recognition technology, which companies like Microsoft and Apple utilise for device security. However, ethical considerations associated with person re-identification must be considered due to factors such as privacy concerns, potential misuse, model limitations and applications, and inadequate data handling practices.

The creation and distribution of datasets such as ‘Duke MTMC’ exemplify significant ethical breaches in data collection, prioritising surveillance technology at the expense of civil, human, and privacy rights. To create the Duke MTMC dataset, images of citizens were compiled without the explicit consent of the individuals involved, resulting in inadvertently enrolling individuals into a global surveillance network used by foreign defence entities [1]. These individuals are now permanently part of a data pool that supports the expansion of biometric surveillance by governments and corporations. This unauthorised collection and use of personal data highlight the ethical concerns associated with person re-identification tasks due to the severe implications that may occur when legal or military entities access this data, potentially significantly impacting individuals’ freedom and rights [2].

Deploying person re-identification technologies such as device authentication and healthcare management systems can enhance digital security and personal wellbeing, providing robust security measures for devices and ensuring effective patient monitoring in medical settings [3]. However, utilising this technology without consent in various applications such as retail to monitor customer behaviour, transportation hubs to track passenger flow, financial services to verify identity during ATM or branch transactions, and their application in public surveillance may erode public anonymity and privacy. Additionally, surveillance technologies utilised by law enforcement or government agencies without stringent oversight may lead to compromised civil liberties due to model limitations [4]. Model limitations, such as inaccuracies in facial recognition, can result in misidentification with severe repercussions. This may disproportionately affect certain demographic groups, amplifying issues such as discrimination due to model bias [5]. Addressing these limitations is crucial to developing fair, unbiased, and reliable technologies that enhance quality of life and improve universal security measures.

REFERENCES

1. Mozur, Paul. (2019). "*One Month, 500,000 Face Scans: How China Is Using A.I. to Profile a Minority*". https://www.nytimes.com/2019/04/14/technology/china-surveillance-artificial-intelligence-racial-profiling.html.
2. Harvey, Adam. Laplace, Jules. (2021). “*Duke Multi-Target Multi-Camera Tracking Dataset*”. https://exposing.ai.
3. Suleski T, Ahmed M, Yang W, Wang E. (2023). A review of multi-factor authentication in the Internet of Healthcare Things. *Digit Health*. doi: 10.1177/20552076231177144.
4. Slobogin C, Brayne S. (2022*).* “Surveillance Technologies and Constitutional Law”. *Annu Rev Criminol*. 6:219-240. doi: 10.1146/annurev-criminol-030421-035102.
5. Almeida D, Shmarko K, Lomas E. (2022). The ethics of facial recognition technologies, surveillance, and accountability in an age of artificial intelligence: a comparative analysis of US, EU, and UK regulatory frameworks*. AI Ethics*. 2022;2(3):377-387. doi: 10.1007/s43681-021-00077-w.

APPENDIX

Table 1:

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓

┃ **Layer (type)** ┃ **Output Shape** ┃ **Param #** ┃

┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩

│ img (InputLayer) │ (None, 64, 32, 3) │ 0 │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ conv2d (Conv2D) │ (None, 64, 32, 16) │ 2,368 │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ batch\_normalization │ (None, 64, 32, 16) │ 64 │

│ (BatchNormalization) │ │ │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ conv2d\_1 (Conv2D) │ (None, 64, 32, 16) │ 12,560 │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ batch\_normalization\_1 │ (None, 64, 32, 16) │ 64 │

│ (BatchNormalization) │ │ │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ max\_pooling2d (MaxPooling2D) │ (None, 32, 16, 16) │ 0 │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ conv2d\_2 (Conv2D) │ (None, 32, 16, 32) │ 12,832 │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ batch\_normalization\_2 │ (None, 32, 16, 32) │ 128 │

│ (BatchNormalization) │ │ │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ conv2d\_3 (Conv2D) │ (None, 32, 16, 32) │ 25,632 │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ batch\_normalization\_3 │ (None, 32, 16, 32) │ 128 │

│ (BatchNormalization) │ │ │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ max\_pooling2d\_1 (MaxPooling2D) │ (None, 16, 8, 32) │ 0 │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ conv2d\_4 (Conv2D) │ (None, 16, 8, 64) │ 18,496 │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ batch\_normalization\_4 │ (None, 16, 8, 64) │ 256 │

│ (BatchNormalization) │ │ │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ conv2d\_5 (Conv2D) │ (None, 16, 8, 64) │ 36,928 │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ batch\_normalization\_5 │ (None, 16, 8, 64) │ 256 │

│ (BatchNormalization) │ │ │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ max\_pooling2d\_2 (MaxPooling2D) │ (None, 5, 2, 64) │ 0 │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ flatten (Flatten) │ (None, 640) │ 0 │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ dense (Dense) │ (None, 301) │ 192,941 │

├─────────────────────────────────┼────────────────────────┼───────────────┤

│ lambda (Lambda) │ (None, 301) │ 0 │

└─────────────────────────────────┴────────────────────────┴───────────────┘

**Total params:** 302,653 (1.15 MB)

**Trainable params:** 302,205 (1.15 MB)

**Non-trainable params:** 448 (1.75 KB)