applied sciences

MDPI

Article

Transformers in Pedestrian Image Retrieval and Person Re-Identification in a Multi-Camera Surveillance System

Muhammad Tahir [1,*,†] **and Saeed Anwar** [2,3,4,†]

[1] College of Computing and Informatics, Saudi Electronic University, Riyadh 11673, Saudi Arabia
[2] Data61-Commonwealth Scientific and Industrial Research Organization(CSIRO), Clayton South, VIC 3169, Australia; saeed.anwar@csiro.au
[3] College of Engineering and Computer Science, Australian National University, Canberra, ACT 2601, Australia
[4] School of Computer Science, University of Technology Sydney, Sydney, NSW 2007, Australia
* Correspondence: m.tahir@seu.edu.sa
† Both the authors contributed equally to this work.

Abstract: Person Re-Identification is an essential task in computer vision, particularly in surveillance applications. The aim is to identify a person based on an input image from surveillance photographs in various scenarios. Most Person re-ID techniques utilize Convolutional Neural Networks (CNNs); however, Vision Transformers are replacing pure CNNs for various computer vision tasks such as object recognition, classification, etc. The vision transformers contain information about local regions of the image. The current techniques take this advantage to improve the accuracy of the tasks underhand. We propose to use the vision transformers in conjunction with vanilla CNN models to investigate the true strength of transformers in person re-identification. We employ three backbones with different combinations of vision transformers on two benchmark datasets. The overall performance of the backbones increased, showing the importance of vision transformers. We provide ablation studies and show the importance of various components of the vision transformers in re-identification tasks.

Keywords: vision transformers; deep learning; re-ID; image retrieval; multi-camera surveillance system; pedestrian identification; person identification

Citation: Tahir, M.; Anwar, S. Transformers in Pedestrian Image Retrieval and Person Re-Identification in a Multi-Camera Surveillance System. *Appl. Sci.* **2021**, *11*, 9197. https://doi.org/10.3390/app11199197

Academic Editor: Daniel Paternain

Received: 14 August 2021
Accepted: 29 September 2021
Published: 2 October 2021

1. Introduction

Person re-identification, abbreviated as "re-ID" primarily focuses on identification and recognition of a person re-appearing in multiple views captured across various cameras in the same surveillance system [1,2]. Person re-ID is crucial to the retrieval of suspects in surveillance camera networks during criminal investigations. Moreover, it is critical in the search for lost people in huge crowds. The mentioned vital applications are indispensable for enhanced public safety and security [3,4]. As a result, the person re-ID problem in computer vision and machine learning has been receiving growing importance and attention in the recent past [5–9].

Typically, a query image is provided to a person re-ID system that searches through a database and returns the matched ones if available. The concept is depicted in Figure 1. In a distributed surveillance system, multiple cameras are capturing the same person from different poses with distinct backgrounds, which may lead to different orientations of the same person [1,10]. Similarly, people's appearance under different lighting and illumination conditions may also affect the learning capability of re-ID systems [11]. Such situations may lead to degraded performance in estimating the similarity between query and candidate images. In order to enhance the image retrieval task for various computer vision applications including person re-ID, Wieczorek et al. [12] formalized a novel Centroid Triplet Loss function.

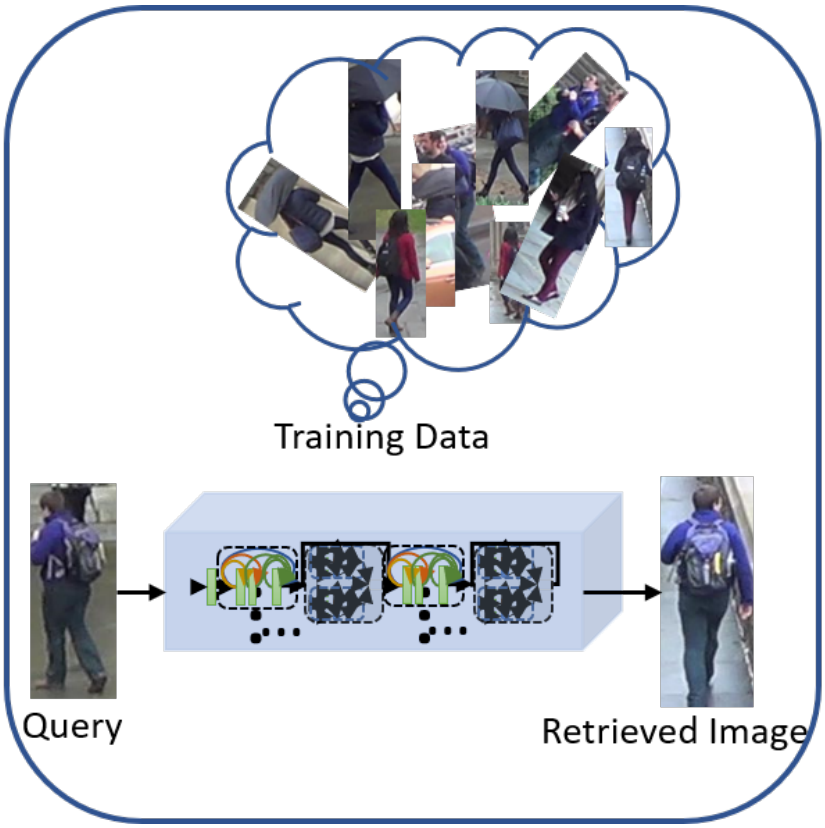


Figure 1. The query image is to be re-identified through the trained model. The training data shows the sample images of the dataset where pictures captured across multiple cameras are stored.

In computer vision and machine learning, person re-ID is regarded as a problem due to the images captured under different lighting conditions, diverse backgrounds, and varying camera views, resulting in considerable intra-class variations. Therefore, the development of robust and stable feature representation methods for re-ID systems has been the center of many researchers' attention in the field.

The development of effective re-identification systems requires the extraction of discriminative features from pedestrian images. However, complications emerging from complex views in a surveillance system limit the learning capability of re-ID systems. This article aims to develop a deep architecture capable of learning discriminative features, specifically using transformers. In other words, we aim to integrate the transformers in the traditional models and person re-id specific models. The purpose is to explore whether the transformers help improve the performance. We will also look into other aspects such as the effect of running time, number of parameters, computational cost, etc.

2. Related Work

Deep networks have been playing a tremendous role in feature representation for person re-identification tasks. In a recent survey, Ye et al. [13] have identified closed-world and open-world approaches to the person re-identification research where closed-world refers to the research conducted under various assumptions in controlled environment and open-world focuses on real-world applications confronting real-world challenges. They also proposed a baseline that demonstrated nearly state-of-the-art performance.

Many researchers have focused on introducing novel architectures to address the problem [9,14–21]. In this section, we briefly discuss some recent deep architectures.

Zhang et al. [22] have developed Feature Pyramid Branch, composed of a global branch and a feature pyramid branch where the global branch uses ResNet50 [23] as the backbone network excluding the final down-sampling procedure of layer 4. The pyramid structure allows feature extraction at various levels and integrates them subsequently for

a person re-ID task. Self-attention modules have also been incorporated into different layers of the proposed network. The authors utilized the consolidated form of triplet loss, cross-entropy loss, and correlation loss.

Sharma et al. [24] developed Locally Aware Transformer for re-ID tasks that is composed of two separate but interconnected networks namely, a backbone network and a locally aware network where the backbone network is the ViT vision transformer [25]. The local and global tokens generated by ViT vision transformer are combined to produce globally enhanced local tokens that are subsequently fed into the locally aware network where the classification results are obtained using the ensemble technique.

Yunpeng [26] developed a multi-modal data learning algorithm for person re-ID tasks that computes global and local homogeneous transformations as well as their combination. The proposed method is capable of learning a relationship between modalities in particular scenarios of person re-ID tasks.

Wang and Zhang [27] devised a multi-label classification algorithm to deal with person re-ID as an unsupervised learning task. ResNet50 is used as a backbone network that is initialized with pre-trained ImageNet parameters. The multi-class labels are predicted iteratively where multi-label classification loss is used to update the network. Initialized with single-class labels, the updated network is adaptively used for predicting multi-class labels that is achieved with memory-based positive label prediction using visual similarity and cycle consistency.

Shu et al. [28] constructed a new large-scale person re-ID dataset LaST that is exploited by ResNet50 network initialized with pre-trained ImageNet parameters. The performance of 14 re-ID algorithms has been evaluated using the new dataset. LaST instances exhibit more realistic scenarios and cover larger spatial and temporal spans. The authors demonstrated the significance of LaST towards a more generalized solution for person re-ID tasks.

Jin et al. [29] proposed CNN based person re-ID model that takes a single image as input rather than a pair or triplet of images for the re-identification task. The proposed model consists of a feature reweighting (FRW) layer, which reweights the input vector from the last fully connected layer to improve the embedding. The final output is obtained by applying identification loss in conjunction with the center loss to re-identify a query image.

Similarly, Xiao et al. [30] utilized ResNet50 [23] as the backbone to learning pedestrian detection and person re-identification jointly using a single convolutional neural network where Detection is performed by pedestrian proposal net whereas re-identification is achieved through identification net. First, the base CNN model's stem part is used to generate convolutional feature maps from the entire gallery image. Next, the pedestrian proposal net utilizes these feature maps to detect candidate pedestrians. Soft-max classifier and linear regression are used to predict actual pedestrians and their locations, respectively. The actual pedestrians are then provided to the identification net for feature extraction. A novel Online Instance Matching loss function is incorporated to guide the training of the re-identification net. The people in the gallery are ranked according to their distances from the target person. The authors collected and annotated a large-scale dataset from streets via hand-held cameras as well as movies. The performance of Xiao et al. [30] outperformed the preceding set of algorithms.

Inspired by the advantages of Siamese architecture [31], Ge et al. [32] developed a Feature Distilling Generative Adversarial Network (FD-GAN), which is composed of image encoder, image generator, identity verifier, and identity as well as pose adversarial discriminators. First, the image encoder, employing ResNet50 [23], extracts features from the input image at each branch. Next, the identity verifier monitors the learning process for the re-identification task. It determines the similarity of the input images. The identity discriminator is designed to retain identity-related information, whereas the pose discriminator is trained to eradicate pose-related information from the extracted features. Similarly, the fundamental concept of FD-GAN is to learn identity-related features while restricting

the system not to rely on pose-related features. The authors utilized cross-entropy loss, adversarial losses of identity and pose discriminators, and reconstruction loss.

Multiple Granularity Network (MGN) [33] is a deep architecture consisting of three independent branches (subnets) with ResNet50 [23] as a backbone. The branches are named based on the type of features it learns, i.e., the upper branch is called Global Branch as it captures global features, whereas the lower two branches learn local features from the input image. During the parts-based feature extraction, the input image is split into three horizontal stripes. The authors employed softmax and triplet loss functions for classification and metric learning, respectively.

Zhong et al. [34] proposed an exemplar memory-based deep network that utilizes pre-trained ResNet50 [23] as backbone. The authors added a fully connected layer after the pooling-5 layer of ResNet50 followed by batch normalization, ReLU activation, Dropout, and two more components. The first component performs supervised learning from labeled data, whereas the second component performs invariance learning from unlabelled data. The exemplar-memory concept is introduced to store the output features of a fully connected layer for all previously seen images from the target domain. Exemplar-memory helps achieve generalization capability concerning exemplar-invariance, camera-invariance, and neighborhood-invariance of the target domain images. The proposed deep network outperformed the existing state-of-the-art approaches based on unsupervised domain adaptation.

The recently-reveloped, Omni-scale network (OSNet) [35] explicitly learns multi-scale features at each layer of the network. The basic building blocks of the network contain multiple convolutional streams capable of extracting features at various scales. The resulting feature maps are combined through a unified aggregation gate. The authors demonstrated that features extracted from local regions and the entire body are equally important for the person's re-ID task. The OSNet [35] performance has been analyzed on six datasets. The results validated its superiority over existing state-of-the-art techniques. Although current methods yield state-of-the-art results, we aim to employ the recently becoming popular modules that are vision transformers used in many computer vision problems, including object detection, object classification, and many other tasks. For this purpose, we utilize different backbones and integrate vision transformers in them to investigate their strength in re-identification tasks. We propose various architectures based on vision transformers and give their performance as well as many insights.

3. Foundation

In this section, we describe the foundation blocks for our proposed methods which consist of Residual Network [23], Dense Network [36], PCB [37], and Transformers [38] to make the article self-contained.

3.1. Residual Network

He et al. [23] proposed the residual networks with identity shortcuts, famously known as skip connections. Identity shortcuts aim to propagate the gradient signal back without being vanishing for deep networks. Theoretically, the identity shortcuts "skip" over all layers of the network, reaching the initial layers to learn the representations. This approach helps learn the offset due to the features' summation at the end of each module; hence, no need to realize the complete features representation by the network. The skip connections help to achieve robust and successful training of deep architectures, which were not possible previously. Figure 4a shows a simple residual architecture of ResNet [23].

3.2. Dense Network

The dense network is presented by Huang et al. [36], where the aim is to provide all the preceding features information to the current convolutional layer in the same block. This technique helps propagate the gradients with ease. Figure 5a shows an overview of the dense network, where all the previous layers' outputs are concatenated and provided

to the current layer. This architecture is different from than residual network, where only a single skip connection is used.

3.3. PCB

Part-based Convolutional Baseline (PCB), proposed by [37], constructs convolutional features from multiple part-level features while applying a uniform partitioning approach on the convolutional layer without explicitly dividing the input image. Any existing image classification network excluding the hidden fully connected layers can be adopted as a backbone to build PCB [37]. The performance of PCB is determined by many essential parameters, including the input image size, the tensor spatial size, and the number of pooled column vectors and enhances its performance by increasing the size of the tensor in the backbone network that is achieved by eradicating the operation of final spatial down-sampling.

3.4. Transformers

The integral components of transformers are (i) self-attention and (ii) multi-headed attention, which is described below.

- **Self-Attention:** The self-attention estimates the significance of one item with others, explicitly modeling the interactions among them for structured prediction, updating each component via global information aggregation from the entire input sequence as shown in Figure 2. Consider a sequence of n items with d embedding dimension i.e., $X = \{x_1, x_2, \cdots, x_n\} \in R^{n \times d}$ then the aim is to capture all the interaction, encoding each entity in terms of the global contextual information by three learnable weight matrices, including *Keys* ($W^K \in R^{n \times d_k}$), *Queries* ($W^Q \in R^{n \times d_q}$) and *Values* ($W^V \in R^{n \times d_v}$), then projecting X on the mentioned matrices to obtain $K = XW^K$, $Q = XW^Q$, and $V = XW^V$ as

$$S_a = softmax\left(\frac{QK^T}{\sqrt{d_q}}\right)V \tag{1}$$

  Here $S_a \in R^{n \times d_v}$ is the self-attention layer's output achieved by computing the dot-product of the query with all keys for a given item; furthermore, softmax is applied to get the normalized attention scores where individual items become the weighted sum of all items. It is to be noted that the attention scores provide weights.
- **Multi-headed Attention:** The multi-head attention shown in Figure 3 is composed of multiple self-attention modules to capture multiple complex relationships between various items in a sequence, where each modules learns the weight matrices W_i^Q, W_i^K, W_i^V, and $i = [0, \cdots, (h-1)]$. At the end of multi-head attention, the h self-attention modules are concatenated $[S_{a0}, S_{a1}, \cdots, Sa_{(h-1)}] \in R^{n \times h \cdot d_v}$ and then projected onto a $W \in R^{h \cdot d_v \times d}$ weight matrix.

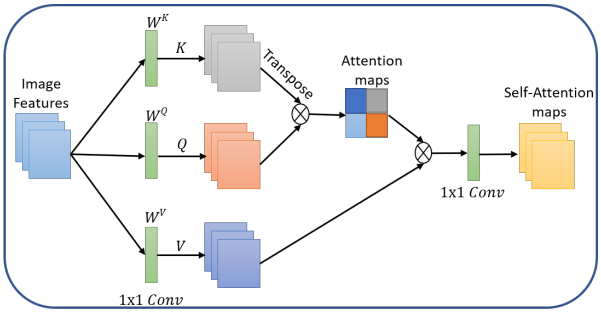


Figure 2. Self-Attention: The convolutional features that are key, query and value are computed. The attention is calculated next and applied to reweight the values. An output projection is employed to obtain output features of the same size as the input.

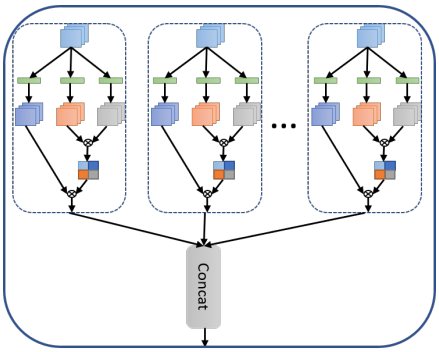


Figure 3. Multi-headed Self-Attention: The self-attention is applied to the same features and then concatenated.

4. Proposed Architectures

We propose three transformer-based networks including *Residual Transformer*, *Dense Transformer* and *PCB Transformer* as follows.

4.1. Residual Transformer

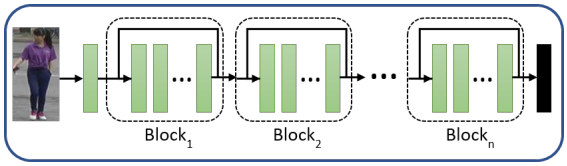
The Residual Transformers (RTr) exploits the building blocks of residual network and Multi-head attention. We propose multiple architectures of the residual network by incorporating different numbers of transformers at various locations in the model. Figure 4b shows the locations where the transformers are included. Suppose we employ a single transformer with h self-attention modules then

$$M_1 = [S_{a0}, S_{a1}, \cdots, Sa_{(h-1)}]. \tag{2}$$

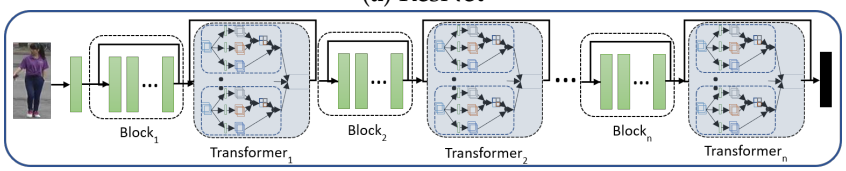
after the first block of the residual network, the mentioned transformer is incorporated as

$$\begin{aligned} \tilde{f} &= \phi_1(f), \\ y_{M_1} &= M_1(\tilde{f}) + \tilde{f}, \\ RTr_{(L_1 M_1 h_4)} &= FC(\phi_4(\phi_3(\phi_2(y_{M_1})))), \end{aligned} \tag{3}$$

where f are the features, and the input to the block (ϕ) of the residual network, the subscript of ϕ represents the block number. The model is termed as $RTr_{(L_1 M_1 h_4)}$ because having only one transformer (M_1) having four heads (h_4), after first level (L_1). We present seven variants of the residual transformers based on the number of transformers and multi-attention heads after each level (blocks) in the residual network.



(**a**) ResNet



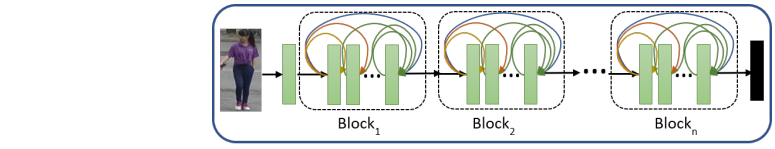
(**b**) Residual Transformer

Figure 4. The architecture of (**a**) ResNet (baseline) and (**b**) Residual Transformers. The fundamental difference between the baseline and Residual Transformers (RTr) is that the transformers after each block is integrated. The number of transformer modules (M), heads (h), and integration after the levels (L) depends on the variant of the Residual Transformer.

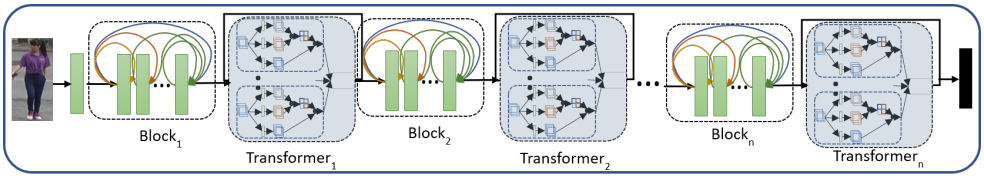
4.2. Dense Transformer

The Dense Transformers (DTr) employs the building blocks of dense networks and transformers discussed earlier in Section 3.2. Let us consider that the features $\tilde{f}$ are the output of the final dense block ψ_f before the fully connected (FC) layer; then we incorporate the transformers as shown in Figure 5b and can be represented as

$$\begin{aligned} \tilde{f} &= \psi_f(f), \\ y_{M_1} &= M_1(\tilde{f}) + \tilde{f}, \\ DTr_{(M_1 h_4)} &= FC(y_{M_1}), \end{aligned} \tag{4}$$



(**a**) DenseNet



(**b**) Dense Transformer

Figure 5. The structure of DenseNet (**a**) and Dense Transformer (**b**). The transformer modules are kept at the network's end instead of each block in the Dense Transformer.

4.3. PCB Transformer

The part-based convolutional baseline uses the residual network as a backbone. We take the same path where the backbone is modified to accommodate the transformers in the architecture as shown in Figure 6. The transformers are added after the blocks as mentioned earlier in *Residual Transformers*. It should be noted that the part-based convolutions are only used in training and removed from the model during the testing phase; hence, transformers can only be employed in the backbone.

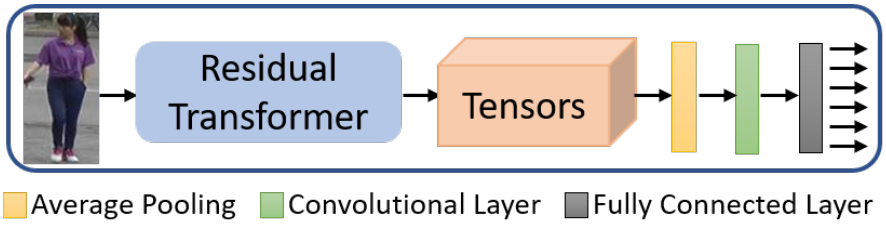


Figure 6. The network structure of the PCB Transformer, which employs the Residual Transformer as a backbone.

5. Experimental Results

5.1. Setup

Datasets: We have used two benchmark re-ID datasets including Market-1501 [39] and DukeMTMCreID [40]; the brief descriptions of these datasets are given below

- *Market-1501* (http://zheng-lab.cecs.anu.edu.au/Project/project_reid.html, accessed on 18 August 2020) dataset [39] is developed by employing six cameras, including one low and five high-resolution cameras outside a supermarket at Tsinghua University where the field-of-view overlap exists between the different cameras. Market-1501

has 32,668 annotated bounding boxes of 1501 pedestrians. For performing the cross-camera search, each pedestrian is captured by all cameras, while it is ensured that a pedestrian is present in at least two cameras.

- *DukeMTMC-reID* (https://github.com/sxzrt/DukeMTMC-reID_evaluation#download-dataset, accessed on 25 August 2020) dataset is constructed from the DukeMTMC [40] dataset, which consists of high-resolution videos acquired by eight cameras with pedestrian annotated bounding boxes. In [40], the pedestrian images are cropped after each 120th frame, yielding 1812 identities having 36,411 bounding boxes. Only 702 IDs are select for training, and 702 IDs are selected for testing, making sure that the pedestrians appear in more than two cameras.

Baselines: We compare against three baselines: ResNet50 [23], DenseNet121 [36] and PCB [37]. These methods are fine-tuned using the benchmark datasets, and their results are used as baselines.

Evaluation Metrics: Two widely used evaluation metrics are employed to evaluate the person re-ID predictions, including mean Average Precision, "mAP" and Accuracy "Acc". Top-1 accuracy is expressed as Rank-1 (R@1), the conventional accuracy where the model outputs the highest probability for the input identity. Top-5 accuracy represented as Rank-5 (R@5) means that any of the five highest probability identities must match the ground truth identity, and top-10 accuracy (R@10) is where the ground truth is present in the top 10 probabilities.

Implementation Details: We use the pre-train weights of ImageNet [41] for convolutional layers and set the batch size to be 16 for training the proposed models with 59 epochs. Stochastic gradient descent (SGD) optimizes the pre-trained model with a momentum of 0.9 and a base learning rate of 0.02, halved after every 20 epochs. We train our proposed models using the PyTorch framework on a PC with V100 GPUs. The time duration for each model varied based on the number of transformers employed. The input size of the image is 256×128 for Residual-Transformer and Dense-Transformer while 384×192 for PCB-Transformer. The minimum number of transformers used in the base model is one, while the maximum is 20.

Objective Function: The loss function is the conventional cross-entropy. To investigate the transformer's ability whether it can learn the discriminative features, we have not experimented with more losses intentionally.

5.2. Comparisons

Performance of Residual-Transformers: We report the performance of various *Residual-Transformers* against baseline in Table 1, where the results demonstrate the benefits of employing the transformers, consistently achieve the best performance on both Market-1501 and DukeMTMC datasets. Specifically, $RTr_{(L_4M_1h_4)}$ achieves the best results around 3.41% and 5.77% for top-1 accuracy (R@1) and mean average precision (mAP), respectively on Market-1501. Similarly, the lowest-performing is $RTr_{(L_1M_3h_4)}$ on Market-1501 which still gets a considerable boost of 1.75% and 3.41% for R@1 and mAP, respectively. Moreover, the accuracy is 5.07% and 3.01% for the highest and lowest for $RTr_{(L_4M_1h_4)}$ and $RTr_{(L_4M_1h_4)}$, respectively on DukeMTMC dataset. This further demonstrates the superior performance of the transformer frameworks integrated into the residual networks.

Performance of Dense-Transformers: Table 2 shows the performance of *Dense-Transformer*. The best performance on Market-1501 is 1.19% (R@1) and 2.99% (mAP) more than baseline achieved by $DTr_{(L_1M_{10}h_4)}$ while for DukeMTMC the increase is 1.34% (R@1) and 1.43% (mAP) by $DTr_{(L_1M_1h_4)}$. The lowest increase is about 0.36% for Market-1501 while for DukeMTMC some of the *Dense-Transformers* obtain less performance.

PCB-Transformers Performance: As a last quantitative comparison, we provide the results of *PCB-Transformers* in Table 3. The performance of the PCB-Transformers is very limited, although it uses the residual network as a backbone. The reason may be that the PCB [37] applies different strategies such as employing triplet loss and training the model for more epochs i.e., 120. The learning rate is dropped by half every 10 epochs between

60–90 epochs. Furthermore, the transformers may also have limited performance due to limited data available for more complex methods; hence, training the PCB-Transformers with ImageNet [41] and then fine-tuning with re-ID may lead to improved performance.

Table 1. Residual Transformers (RTr) performance against baseline (ResNet50) trained and evaluated on Market-1501 [39] and DukeMTMC [40] datasets. The best results are in bold, and "-" represents that the network did not converge in the given number of epochs. The "Levels" represent the presence of transformers after that block, while the subscript of "M" means the number of transformers after each block. The number of heads "h" is four throughout the experiments.

	Level-1 (L_1)				**Level-2 (L_2)**				**Level-3 (L_3)**				**Level-4 (L_4)**			
	R@1	**R@5**	**R@10**	**mAP**	**R@1**	**R@5**	**R@10**	**mAP**	**R@1**	**R@5**	**R@10**	**mAP**	**R@1**	**R@5**	**R@10**	**mAP**
Market-1501 Dataset																
ResNet50	87.65	94.69	96.85	71.15												
M_1	90.86	96.44	97.92	74.99	90.77	96.59	97.92	75.61	90.38	95.96	97.71	74.80	**91.06**	**96.59**	**97.92**	**76.92**
M_2	90.08	96.29	97.63	74.78	90.97	96.44	97.83	76.63	90.20	96.35	97.95	76.00	-	-	-	-
M_3	90.26	96.41	97.65	75.14	90.26	96.23	97.45	75.59	90.53	96.08	97.42	74.87	89.85	96.20	97.74	74.69
M_4	89.70	96.20	97.48	74.21	89.40	96.50	97.77	74.56	89.67	96.41	97.95	74.74	-	-	-	-
M_5	90.20	96.32	97.92	74.42	90.50	96.50	97.68	75.02	90.17	96.38	97.92	74.36	89.61	96.11	97.83	74.44
DukeMTMC Dataset																
ResNet50	77.74	87.84	91.34	60.65												
M_1	81.33	90.62	93.04	64.35	81.46	90.66	93.04	65.10	81.82	90.62	93.36	65.51	**82.81**	**91.20**	**93.36**	**66.34**
M_2	81.19	91.29	93.90	64.68	81.60	90.98	93.58	64.56	82.41	90.93	93.18	65.70	-	-	-	-
M_3	80.75	90.75	93.45	64.54	81.64	91.11	93.94	65.20	81.55	91.07	93.54	65.82	-	-	-	-
M_4	81.37	90.66	93.09	65.27	82.14	90.98	93.13	65.67	81.87	90.35	93.27	65.50	81.19	90.17	93.40	65.80
M_5	80.88	90.53	92.86	64.86	81.15	89.81	92.64	64.20	81.51	90.40	93.36	65.08	81.78	90.80	93.81	65.33

Table 2. The performance of the Dense Transformers (DTr) and baseline (DenseNet121) for Market-1501 [39] and DukeMTMC [40] datasets. The best results are in bold, and "-" represents that the network did not converge in the given number of epochs. The subscripts of "L", "M" and "h" represent the presence of transformers after that block, the number of transformers after each block, and the number of heads in the model, respectively.

R@1	**R@5**	**R@10**	**mAP**	**R@1**	**R@5**	**R@10**	**mAP**	**R@1**	**R@5**	**R@10**	**mAP**	**R@1**	**R@5**	**R@10**	**mAP**
Market-1501 Dataset															
DenseNet				$DTr_{(L_1M_1h_4)}$				$DTr_{(L_1M_2h_4)}$				$DTr_{(L_1M_3h_4)}$			
90.02	95.99	97.42	73.74	90.94	97.00	97.95	75.68	90.47	96.59	97.71	76.28	90.56	96.59	97.89	75.51
$DTr_{(L_1M_4h_4)}$				$DTr_{(L_1M_5h_4)}$				$DTr_{(L_1M_6h_4)}$				$DTr_{(L_1M_7h_4)}$			
90.65	96.70	97.74	74.43	90.86	96.32	97.62	75.73	90.68	96.79	98.04	75.66	90.88	96.82	97.83	75.31
$DTr_{(L_1M_8h_4)}$				$DTr_{(L_1M_9h_4)}$				$DTr_{(L_1M_{10}h_4)}$				$DTr_{(L_1M_{11}h_4)}$			
90.91	96.50	97.89	76.29	91.06	96.79	98.04	75.37	**91.21**	**97.00**	**98.07**	**76.73**	90.38	96.59	97.89	74.97
$DTr_{(L_1M_{12}h_4)}$				$DTr_{(L_1M_{13}h_4)}$				$DTr_{(L_1M_{14}h_4)}$				$DTr_{(L_1M_{15}h_4)}$			
91.00	96.50	97.62	75.47	90.86	96.82	98.10	75.76	90.62	96.59	97.89	74.95	90.56	96.29	97.62	75.33
DukeMTMC Dataset															
DenseNet				$DTr_{(L_1M_1h_4)}$				$DTr_{(L_1M_2h_4)}$				$DTr_{(L_1M_3h_4)}$			
82.05	90.31	92.82	64.61	**83.39**	**91.43**	**93.85**	**66.04**	83.21	91.79	94.30	66.84	82.99	91.61	93.54	66.92
$DTr_{(L_1M_4h_4)}$				$DTr_{(L_1M_5h_4)}$				$DTr_{(L_1M_6h_4)}$				$DTr_{(L_1M_7h_4)}$			
81.78	91.25	93.67	65.76	82.09	91.16	93.54	66.32	83.08	91.52	93.58	65.79	82.50	91.56	94.21	65.87
$DTr_{(L_1M_8h_4)}$				$DTr_{(L_1M_9h_4)}$				$DTr_{(L_1M_{10}h_4)}$				$DTr_{(L_1M_{11}h_4)}$			
82.63	91.34	93.81	65.30	81.87	90.48	93.36	66.45	82.18	90.80	93.49	66.23	81.82	91.02	93.31	65.93
$DTr_{(L_1M_{12}h_4)}$				$DTr_{(L_1M_{13}h_4)}$				$DTr_{(L_1M_{14}h_4)}$				$DTr_{(L_1M_{15}h_4)}$			
81.96	91.38	94.21	65.99	81.60	90.48	93.18	64.40	82.59	90.62	93.67	65.79	82.85	91.07	94.03	66.54

Table 3. The performance of the PCB Transformers (PCBTr) and baseline (PCB) for Market-1501 [39] datasets. The best results are in bold. The subscripts of "L", "M" and "h" represent the presence of transformers after that block, the number of transformers after each block, and the number of heads in the model, respectively.

Methods	R@1	R@5	R@10	mAP
PCB	**92.64**	-	-	**77.47**
$PCBTr_{(L_1M_1h_4)}$	86.58	94.42	96.26	66.00
$PCBTr_{(L_1M_2h_4)}$	90.08	95.72	97.15	71.95
$PCBTr_{(L_2M_2h_4)}$	88.00	95.25	96.67	68.23
$PCBTr_{(L_3M_2h_4)}$	72.36	86.67	90.53	42.64

5.3. Ablation Studies

In this section, we provide some of the aspects of the proposed architectures.

Influence on Training Time: One of the critical components of any model is the training time. Table 4 shows the comparison between the training time for the baseline network and various RTr models on the Market-1501 dataset. It is evident that the training time increases drastically (from 55min to 139 min) even employing two transformers in the model. Moreover, the training becomes much slower when the number of transformers increases as the training time is directly proportional to the number of transformers.

Table 4. Comparison on the Market dataset between the Residual Transformers in terms of training time (in minutes) and the number of parameters (in Millions).

	Baseline	$RTr_{(L_1M_2h_4)}$	$RTr_{(L_2M_2h_4)}$	$RTr_{(L_3M_2h_4)}$	$RTr_{(L_4M_2h_4)}$	$RTr_{(L_1M_3h_4)}$	$RTr_{(L_2M_3h_4)}$	$RTr_{(L_3M_3h_4)}$	$RTr_{(L_4M_3h_4)}$
Time	55 min	139 min	154 min	164 min	178 min	152 min	197 min	212 min	234 min
Parameters	24.94 M	25.34 M	26.91 M	33.21 M	58.39 M	25.34 M	27.90 M	37.35 M	75.11 M

Increase in Number of Parameters: The number of parameters is also increased when transformers are integrated into the baselines, as shown in Table 4 (second row). In this case, the baseline architecture i.e., *ResNet*50 has about 24.94 M parameters compared to 25.34 M parameters with only two transformers i.e., $RTr_{(L_1M_2h_4)}$. The number of parameters significantly becomes more when the transformers number increases i.e., $RTr_{(L_4M_3h_4)}$ having 75.11 M parameters due to 12 transformer modules.

Effect on the Computational Cost: Compared to the base model, the transformer models adversely affect the computational cost in terms of time and number of parameters. Moreover, it should be noted that the parameters due to transformers and computations required for computing the self-attention in multi-head attention also affect the inference time; hence, need more time for re-identification. Overall, the computational cost increases due to the integration of transformers in base models.

Attention Focus in the Images: In this section, we provide the focus of the transformers against the baselines. Figure 7 provides the focus maps similar to [42,43] with the corresponding original images. Moreover, for each image, three explanation maps are generated via Grad-CAM++ [44] (1st row), Score-CAM [45] (2nd row), and Eigen-CAM [46] (3rd row). Compared to baseline, our proposed transformer architectures focus on the specific details of the persons; for example, in the second row, the baseline focuses on the whole body while most transformers-based methods focus on the detailed specific body parts.

Number of Attention Heads: We also investigate the effect of different numbers of self-attention heads employed in transformers. Table 5 shows the results for 1, 2, 4, and 16 headed self-attentions i.e., h_1, h_2, h_4 *and* h_{16} for Market-1501 integrated across three different levels utilizing three transformer modules while keeping all other training factors constant. Most of the best results are achieved for 4-headed attention; hence, we use h_4 in all our experiments. It should also be noted that for h_{16}, the models failed to converge.

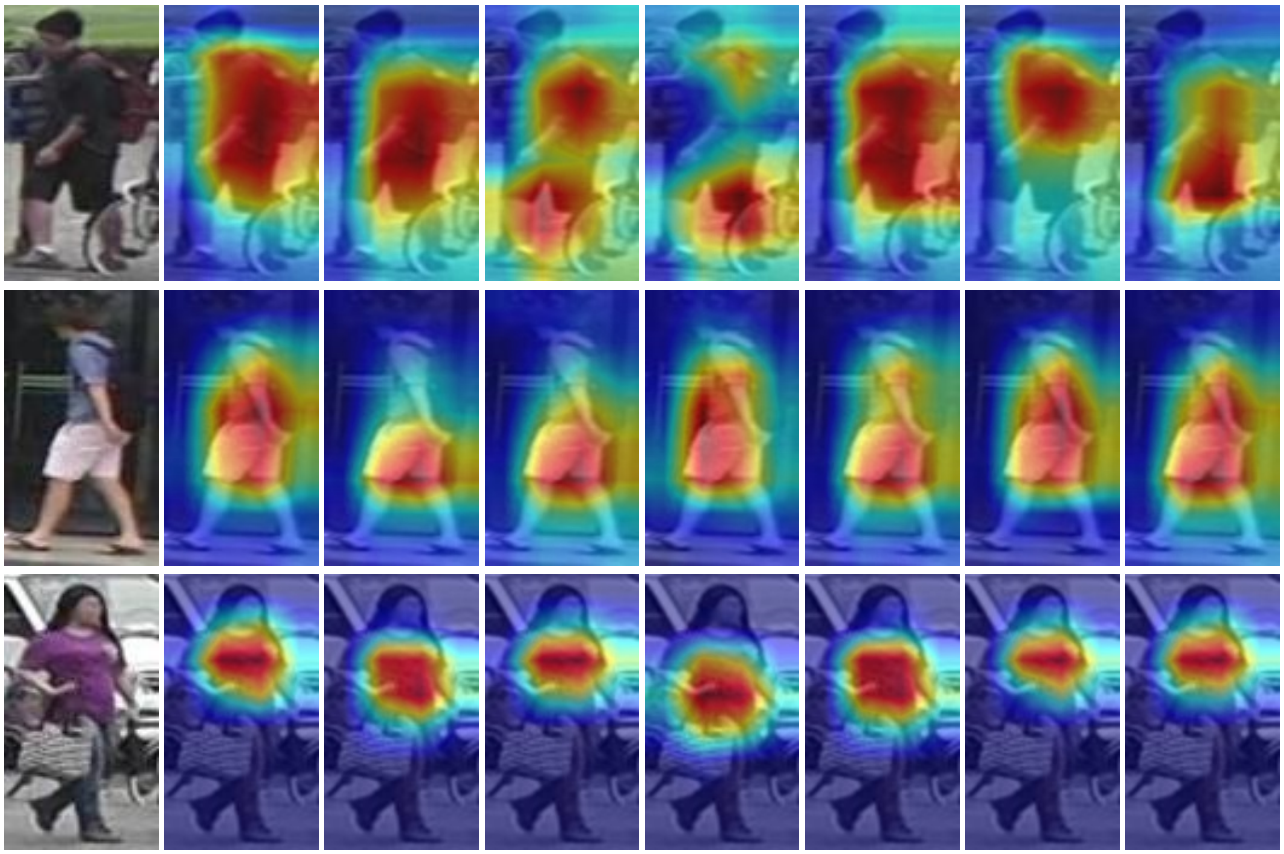


Figure 7. Sample images from the Market-1501 dataset to show the visual explanations of Residual-Transformers against the baseline method. Our proposed architecture focused on the fine-grained details to re-identify. The visual attention maps are generated by baseline and transformer-based architectures using Grad-CAM++ [44] (1st row), Score-CAM [45] (2nd row), and Eigen-CAM [46] (3rd row).

Table 5. The effect of the number of heads on various Residual transformers trained and evaluated on Market-1501 dataset. The worst performance is given when the number of heads is 16.

	Level-1 (L_1)				**Level-2 (L_2)**				**Level-3 (L_3)**			
	R@1	**R@5**	**R@10**	**mAP**	**R@1**	**R@5**	**R@10**	**mAP**	**R@1**	**R@5**	**R@10**	**mAP**
h_1	89.40	95.78	97.65	74.41	90.23	96.11	97.65	75.29	90.74	96.41	97.83	75.47
h_2	89.46	96.32	97.57	74.33	90.32	96.47	97.65	75.80	90.59	96.41	97.65	75.45
h_4	90.26	96.41	97.65	75.14	90.26	96.23	97.45	75.59	90.53	96.08	97.42	74.87
h_{16}	0.06	0.06	0.06	0.20	0.06	0.06	0.06	0.20	0.06	0.06	0.06	0.20

The Impact of Using Different Number of Transformers: Another essential component to determine is the number of transformers required to boost the performance of the baselines. We have incorporated between 1 to 25 transformers for RTr and 1 to 16 transformers in DTr as shown in Tables 1 and 2. There is no specific number of transformers that gives the highest performance on all datasets over all the baselines. However, it can be seen that single transformers provide a considerable improvement over baselines.

Effect of Transformers Locations: The role of integrating Transformers into the model is essential. We have placed the transformers in *RTr* after blocks while in *DTr* at the network's end before the classification layer. In *RTr*, the best results are obtained when a single transformer is placed after each in the baseline for both datasets, as shown in Table 1. Similarly, the best performance is achieved using a single transformer for DukeMTMC and 10 transformers for Market-1501 datasets, as shown in Table 2. However, it should be noted

that irrespective of the integration location of transformers, improvement is achieved in most cases.

6. Conclusions

In this article, we proposed using transformers in many backbones for a person's image retrieval and re-identification in multi-camera surveillance systems. We compared their effects via various metrics and datasets and provide analysis on the performance. We summarized the impact of this mechanism in person re-identification in terms of time consumed during training, increase in the number of parameters, the effect of the number of heads, number of modules, and integration location. We provided 4, 15, and 25 variants of PCB, Residual and Dense transformers, respectively, on two benchmark datasets. We conclude that transformers improve the performance in most cases at the cost of the number of parameters and longer training times; therefore, efficient transformers are the need of the hour. We hope that our findings will help the community and constitute a baseline for future work.

Author Contributions: Conceptualization, M.T. and S.A.; Funding acquisition, M.T.; Investigation, M.T. and S.A.; Methodology, S.A.; Project administration, M.T.; Resources, M.T.; Validation, S.A.; Writing—original draft, M.T. and S.A.; Writing—review & editing, S.A. All authors have read and agreed to the published version of the manuscript

Funding: The authors extend their appreciation to the Deanship of Scientific Research at Saudi Electronic University, Riyadh, Saudi Arabia for funding this work under grant number 7697-CAI-2019-1-2-r.

Data Availability Statement: The code and models are available at https://github.com/saeed-anwar/TRE-ID.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zheng, L.; Yang, Y.; Hauptmann, A.G. Person re-identification: Past, present and future. *arXiv* **2016**, arXiv:1610.02984.
2. Bai, X.; Yang, M.; Huang, T.; Dou, Z.; Yu, R.; Xu, Y. Deep-person: Learning discriminative deep features for person re-identification. *arXiv* **2017**, arXiv:1711.10658.
3. Yan, Y.; Zhang, Q.; Ni, B.; Zhang, W.; Xu, M.; Yang, X. Learning Context Graph for Person Search. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2153–2162.
4. Bakalos, N.; Voulodimos, A.; Doulamis, N.; Doulamis, A.; Ostfeld, A.; Salomons, E.; Caubet, J.; Jimenez, V.; Li, P. Protecting Water Infrastructure From Cyber and Physical Threats: Using Multimodal Data Fusion and Adaptive Deep Learning to Monitor Critical Systems. *IEEE Signal Process. Mag.* **2019**, *36*, 36–48. [CrossRef]
5. Xu, Y.; Ma, B.; Huang, R.; Lin, L. Person search in a scene by jointly modeling people commonness and person uniqueness. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; ACM: New York, NY, USA, 2014; pp. 937–940.
6. Dai, Z.; Chen, M.; Zhu, S.; Tan, P. Batch feature erasing for person re-identification and beyond. *arXiv* **2018**, arXiv:1811.07130.
7. Huang, H.; Yang, W.; Chen, X.; Zhao, X.; Huang, K.; Lin, J.; Huang, G.; Du, D. EANet: Enhancing Alignment for Cross-Domain Person Re-identification. *arXiv* **2018**, arXiv:1812.11369.
8. Zheng, Z.; Yang, X.; Yu, Z.; Zheng, L.; Yang, Y.; Kautz, J. Joint discriminative and generative learning for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2133–2142.
9. Wang, G.; Lai, J.; Huang, P.; Xie, X. Spatial-temporal person re-identification. In Proceedings of the AAAI Conference on Artificial Intelligence, Hilton Hawaiian Village, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8933–8940.
10. Yang, F.; Yan, K.; Lu, S.; Jia, H.; Xie, X.; Gao, W. Attention driven person re-identification. *Pattern Recognit.* **2019**, *86*, 143–155. [CrossRef]
11. Adaimi, G.; Kreiss, S.; Alahi, A. Rethinking Person Re-Identification with Confidence. *arXiv* **2019**, arXiv:1906.04692.
12. Wieczorek, M.; Rychalska, B.; Dabrowski, J. On the Unreasonable Effectiveness of Centroids in Image Retrieval. *arXiv* **2021**, arXiv:2104.13643.
13. Ye, M.; Shen, J.; Lin, G.; Xiang, T.; Shao, L.; Hoi, S.C. Deep learning for person re-identification: A survey and outlook. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef]
14. Wang, H.; Fan, Y.; Wang, Z.; Jiao, L.; Schiele, B. Parameter-Free Spatial Attention Network for Person Re-Identification. *arXiv* **2018**, arXiv:1811.12150.

15. Wojke, N.; Bewley, A. Deep cosine metric learning for person re-identification. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 748–756.
16. Zhong, Z.; Zheng, L.; Zheng, Z.; Li, S.; Yang, Y. Camera style adaptation for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5157–5166.
17. Zheng, F.; Deng, C.; Sun, X.; Jiang, X.; Guo, X.; Yu, Z.; Huang, F.; Ji, R. Pyramidal Person Re-IDentification via Multi-Loss Dynamic Training. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8506–8514.
18. Luo, H.; Gu, Y.; Liao, X.; Lai, S.; Jiang, W. Bag of tricks and a strong baseline for deep person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019; pp. 1487–1495.
19. Quan, R.; Dong, X.; Wu, Y.; Zhu, L.; Yang, Y. Auto-ReID: Searching for a Part-aware ConvNet for Person Re-Identification. *arXiv* **2019**, arXiv:1903.09776.
20. Ro, Y.; Choi, J.; Jo, D.U.; Heo, B.; Lim, J.; Choi, J.Y. Backbone Can Not be Trained at Once: Rolling Back to Pre-trained Network for Person Re-identification. *arXiv* **2019**, arXiv:1901.06140.
21. Zeng, Z.; Wang, Z.; Wang, Z.; Chuang, Y.Y.; Satoh, S. Illumination-Adaptive Person Re-identification. *arXiv* **2019**, arXiv:1905.04525.
22. Zhang, S.; Yin, Z.; Wu, X.; Wang, K.; Zhou, Q.; Kang, B. FPB: Feature Pyramid Branch for Person Re-Identification. *arXiv* **2021**, arXiv:2108.01901.
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
24. Sharma, C.; Kapil, S.R.; Chapman, D. Person Re-Identification with a Locally Aware Transformer. *arXiv* **2021**, arXiv:2106.03720.
25. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
26. Yunpeng, G. A general multi-modal data learning method for Person Re-identification. *arXiv* **2021**, arXiv:2101.08533.
27. Wang, D.; Zhang, S. Unsupervised person re-identification via multi-label classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10978–10987.
28. Shu, X.; Wang, X.; Zhang, S.; Zhang, X.; Chen, Y.; Li, G.; Tian, Q. Large-Scale Spatio-Temporal Person Re-identification: Algorithm and Benchmark. *arXiv* **2021**, arXiv:2105.15076.
29. Jin, H.; Wang, X.; Liao, S.; Li, S.Z. Deep person re-identification with improved embedding and efficient training. In Proceedings of the 2017 IEEE International Joint Conference on Biometrics (IJCB), Denver, CO, USA, 1–4 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 261–267.
30. Xiao, T.; Li, S.; Wang, B.; Lin, L.; Wang, X. Joint detection and identification feature learning for person search. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July, 2017; pp. 3376–3385.
31. Bromley, J.; Bentz, J.; Bottou, L.; Guyon, I.; LeCun, Y.; Moore, C.; Sackinger, E.; Shah, R. Signature Verification using a "Siamese" Time Delay Neural Network. *Int. J. Pattern Recognit. Artif. Intell.* **1993**, *7*, 669–688. [CrossRef]
32. Ge, Y.; Li, Z.; Zhao, H.; Yin, G.; Yi, S.; Wang, X.; Li, H. FD-GAN: Pose-guided feature distilling GAN for robust person re-identification. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 1230–1241.
33. Wang, G.; Yuan, Y.; Chen, X.; Li, J.; Zhou, X. Learning discriminative features with multiple granularities for person re-identification. In Proceedings of the 2018 ACM Multimedia Conference on Multimedia Conference, Seoul, Korea, 22–26 October 2018; ACM: New York, NY, USA, 2018; pp. 274–282.
34. Zhong, Z.; Zheng, L.; Luo, Z.; Li, S.; Yang, Y. Invariance matters: Exemplar memory for domain adaptive person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 598–607.
35. Zhou, K.; Yang, Y.; Cavallaro, A.; Xiang, T. Omni-Scale Feature Learning for Person Re-Identification. *arXiv* **2019**, arXiv:1905.00953.
36. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
37. Sun, Y.; Zheng, L.; Yang, Y.; Tian, Q.; Wang, S. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 480–496.
38. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
39. Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; Tian, Q. Scalable person re-identification: A benchmark. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1116–1124.
40. Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; Tomasi, C. Performance measures and a data set for multi-target, multi-camera tracking. In Proceedings of the European Conference on Computer Vision, Workshops, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 17–35.

41. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. Imagenet: A large-scale hierarchical image database. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 248–255.
42. Tahir, M.; Anwar, S.; Mian, A. Deep localization of protein structures in fluorescence microscopy images. *arXiv* **2018**, arXiv:1910.04287.
43. Anwar, H.; Anwar, S.; Zambanini, S.; Porikli, F. Deep ancient Roman Republican coin classification via feature fusion and attention. *Pattern Recognit.* **2021**, *114*, 107871. [CrossRef]
44. Chattopadhay, A.; Sarkar, A.; Howlader, P.; Balasubramanian, V.N. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In -Proceedings of the Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 839–847.
45. Wang, H.; Wang, Z.; Du, M.; Yang, F.; Zhang, Z.; Ding, S.; Mardziel, P.; Hu, X. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 24–25.
46. Muhammad, M.B.; Yeasin, M. Eigen-CAM: Class Activation Map using Principal Components. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–7.