Lightweight pyramid residual features with attention for person re-identification

Reza Fuad Rachmadi [a,1,*], I Ketut Eddy Purnama [a,2], Supeno Mardi Susiki Nugroho [a,3]

[a] Dept. of Computer Engineering, Institut Teknologi Sepuluh Nopember , Surabaya, Indonesia
[1] fuad@its.ac.id ; [2] ketut@te.its.ac.id; [3] mardi@its.ac.id
* corresponding author

ARTICLE INFO

Article history
Received August 16, 2021
Revised January 1, 2023
Accepted January 30, 2023
Available online March 31, 2023

Keywords
Lightweight residual network
Pyramid attention network
Person re-identification
Atrous convolution

ABSTRACT

Person re-identification is one of the problems in the computer vision field that aims to retrieve similar human images in some image collections (or galleries). It is very useful for people searching or tracking in a closed environment (like a mall or building). One of the highlighted things on person re-identification problems is that the model is usually designed only for performance instead of performance and computing power consideration, which is applicable for devices with limited computing power. In this paper, we proposed a lightweight residual network with pyramid attention for person re-identification problems. The lightweight residual network adopted from the residual network (ResNet) model used for CIFAR dataset experiments consists of not more than two million parameters. An additional pyramid features extraction network and attention module are added to the network to improve the classifier's performance. We use CPFE (Context-aware Pyramid Features Extraction) network that utilizes atrous convolution with different dilation rates to extract the pyramid features. In addition, two different attention networks are used for the classifier: channel-wise and spatial-based attention networks. The proposed classifier is tested using widely use Market-1501 and DukeMTMC-reID person re-identification datasets. Experiments on Market-1501 and DukeMTMC-reID datasets show that our proposed classifier can perform well and outperform the classifier without CPFE and attention networks. Further investigation and ablation study shows that our proposed classifier has higher information density compared with other person re-identification methods.

1. Introduction

In recent years, artificial intelligence technology (mainly after deep learning introduces) has been applied to many applications, including ASR (Audio Sound Recognition) [1],[2] image or scene classification [3], [4], video analysis [5], games [6], [7] autonomous vehicle [8], [9] and financial sector [10]. Person re-identification is introduced as an image-based person search in a closed environment, such as a mall or building. Usually, the person images were extracted from CCTV cameras placed in several places. The person re-identification process is processing a person's query image and retrieving highly similar person images from the gallery.

To support person re-identification research, several datasets have been proposed by researchers [11]–[14] along with several proposed person re-identification models [15]–[20]. Although several proposed person re-identification models have been produced, one highlighted point of the current person re-identification model is that it focuses only on performance instead of performance and execution time.

The development of lightweight classifiers can help implement person re-identification on limited computing power scenarios, such as IoT applications and edge computing.

One way to increase the performance of a CNN classifier is by utilizing attention [21], [22] and pyramid features [23], [24]. Attention is a network that predicts features map importantness and weighted the final features such that several features will gain more weight than others based on learning data. Another method to improve the classifier's performance is by extracting pyramid features that combine the last layer convolutional features with several middle-layer features to form the final features.

In this paper, we investigated the lightweight residual network with pyramid attention for person re-identification problems. The proposed classifier is adapted from the lightweight residual network (ResNet) classifier originally used for the CIFAR dataset, which is considered a lightweight classifier with not more than 2 million parameters in the classifier. The pyramid attention mechanism [21] is then attached to the classifier to improve the classifier's performance. Our contributions can be listed as follows.

- We investigated the combination of the lightweight ResNet classifier with pyramid attention for person re-identification. Five lightweight ResNet classifiers were used with a maximum of around 2 million parameters.
- We investigated three pyramid attention networks, channel-based, spatial-based, and a combination of those two attention types. Experiments on Market-1501 and DukeMTMC-reID datasets show that the best performance was achieved using a combination of channel and spatial-based attention.
- We perform an ablation study for several settings, including the size of features used to perform the retrieval process, the re-ranking effects, and dataset bias. Experiments using ensemble configuration were also performed to improve the classifier's performance.

The rest of the paper is organized as follows. The proposed classifier and experiments setup are discussed in section 2. The main results and ablation study are discussed in section 3. Finally, we conclude the experiments in the last section.

2. Method

The proposed classifier consists of three parts: the lightweight residual network, CPFE (Context-aware Pyramid Features Extraction) network, and the attention network. The methods were chosen because it is proven can improve the classifier's performance as described in [21]–[24]. In this section, all of those parts will be discussed in detail. The complete diagram of our proposed classifier can be viewed in Fig. 1.

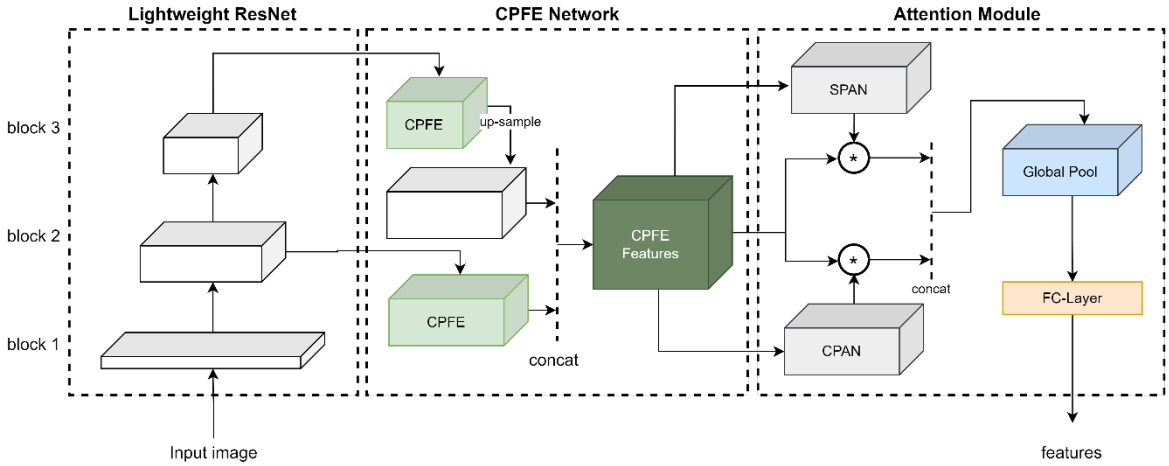

Fig. 1. The Diagram of our proposed classifier consists of three parts: the lightweight residual network with three residual blocks, CPFE network, and attention module.

2.1. Lightweight Residual Network

We use the residual network classifier originally used in CIFAR experiments [25], which works very well for person re-identification [26]. The advantages of the residual network classifier are that the classifier has a low number of parameters which is very suitable for limited resource scenarios. This paper uses two residual networks that produce the highest accuracy, as reported in [26], ResNet-56 and ResNet-110 classifiers. Although the residual network consists of many layers (e.g., 56 and 110 layers), the number of parameters in the classifier is not more than 2 million, which is considered very lightweight. In detail, the residual network consists of three residual blocks in which the output of the last two blocks is used to extract the pyramid features via the CPFE network. After adding the CPFE and attention network, the number of parameters in the classifier increased from 100K to 700K, depending on the attention network type.

2.2. CPFE Network

To utilize more effective feature extraction for classification and/or re-identification problems, the proposed classifier is designed to extract pyramid-based features via the CPFE network. The CPFE network is described in [21], which was originally used for saliency detection problems. Atrous convolution layers using different dilation rates with input from the different residual blocks are used to construct the CPFE network. The atrous convolution process can extract features from different receptive field sizes which match with pyramid features definition. Let $\mathbf{X} \in \Re^{n \times m}$ is an input of the atrous convolution process, $\mathbf{W} \in \Re^{s \times s}$ is the kernel used for the convolution process, and r is the dilation rate, the output of atrous convolution can compute as follows.

$$\mathrm{Y(X)} = \sum_{i,j,k,l} \mathrm{X}[i + (r * k), j + (r * l)] * \mathrm{W}[k, l] \tag{1}$$

he output of the CPFE network is concatenated of four different parallel convolutional layers, one layer with a kernel size of $s = 1$ and dilation rate of $r = 1$; and three other layers with a kernel size of $s = 3$ and dilation rate of $r = \{3, 5, 7\}$.

2.3. Channel-wise Pyramid Attention Network

The output of the convolutional process in CNN classifiers can be interpreted as a channel-wise independent feature extraction process. Each channel of convolutional output represented one semantic feature extraction map, and the importantness of each channel may vary depending on the input and the problem solved by the classifier. The channel-wise pyramid attention network works by weighting each channel based on importantness. We adopted the channel-wise pyramid attention network described by Zhao et al. [21]. Let $\mathbf{f} \in \Re^{c}$ is the unfolded global pooling high-level features with c channels, $\mathbf{W}_1 \in \Re^{c \times (c/2)}$ is the weight of the first fully-connected layer and $\mathbf{W}_1 \in \Re^{(c/2) \times c}$ is the weight of the second fully-connected layer, the channel-wise attention output can be computed as follows

$$\mathbf{a}_{ca} = \sigma(\delta(\mathbf{f} \cdot \mathbf{W}_1) \cdot \mathbf{W}_2 \tag{2}$$

with σ is the sigmoid function and δ is the ReLU activation function. The final features are computed by weighting the features with the output of channel-wise attention module $\mathbf{a}_{ca}$.

$$\hat{\mathbf{F}}_{ca} = \mathbf{F}_c \cdot \mathbf{a}_{ca}[c] \quad \forall c \tag{3}$$

The $\cdot$ operation in equation (3) is scalar multiplication of $\mathbf{F}_c$ features and $c = \{1, 2, \cdots, c_m\}$ with c_m is the number of channels.

2.4. Spatial Pyramid Attention Network

For the spatial pyramid attention network, we also adapted the spatial pyramid attention network described in [21]. Although the spatial pyramid attention network described in [21] was used for saliency detection, our investigation shows that the attention method can also be used for other tasks. Let $\mathbf{X} \in \Re^{w \times h \times c}$ is the input of spatial attention network, $\mathbf{W}_1^g$ is the convolution with a kernel size of $1 \times k$,

and $\mathbf{W}_2^g$ is the convolution with a kernel size of $k \times 1$, the output spatial attention network can be computed as follows.

$$\mathbf{L}_1 = \delta(\delta(\mathbf{X} * \mathbf{W}_1^1) * \mathbf{W}_2^1) \quad (4)$$

$$\mathbf{L}_2 = \delta(\delta(\mathbf{X} * \mathbf{W}_1^2) * \mathbf{W}_2^2) \quad (5)$$

$$\mathbf{A}_{sp} = \sigma(\mathbf{L}_1 + \mathbf{L}_2) \quad (6)$$

with $\mathbf{A}_{sp} \in \Re^{w \times h}$ is the output of the spatial attention network, is the ReLU activation function, and δ is the sigmoid function. The final features computed as follows.

$$\acute{\mathbf{F}}_{sp} = \mathbf{F}_c \circ \mathbf{A}_{sp} \quad \forall c \quad (7)$$

The $\circ$ operation is the element-wise multiplication operation of $\mathbf{F}_c$ features and $c = \{1, 2, \cdots, c_m\}$ with c_m is the number of channels.

2.5. Implementation Details

Our proposed classifier takes input images with a resolution of 64x32. The input resolution was used because it was close to the original LRN input resolution on the CIFAR dataset.

Classifier configuration. We use three different classifier configurations that can be viewed in Fig. 2. In the first version (or v1), we use the multi-loss function (for original LRN and CPFE features) to train the classifier with two parallel heads. The advantages of the first version are that the two parallel head produces an individual loss, improving the classifier faster than other configurations. In addition, the retrieval process will use twice more features compared with other configurations.

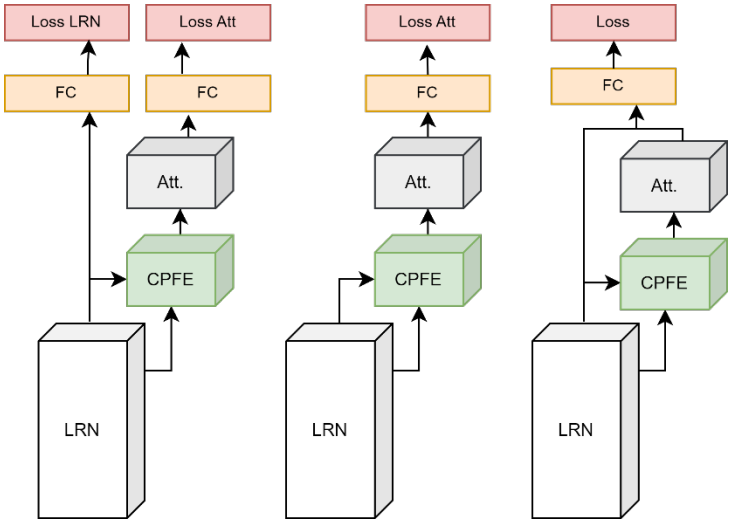

Fig. 2. Three different connectivity between LRN, CPFE, and attention module of our proposed classifier, (a) multi-loss (v1), (b) attention features only (v2), and (c) one loss with combined features (v3).

The second version (or v2) only uses the features produced by CPFE and the attention network. The third version (or v3) combines original features with features produced by CPFE and the attention network. We use the cross-entropy loss function for all configurations and train the classifier for identity identification problems. Let t_k be the *k-th* class of true data label and p_k be the probability score of data being a class k, the cross-entropy loss function can be computed as follows.

$$L_{ce} = -t_k \log(p_k) \quad (8)$$

To create the final loss, we sum up the loss value of two different parallel heads for the first classifier configuration. The last FC block of our proposed classifier consists of two fully connected layers, a fully

connected layer with 256 neurons with a batch normalization layer but without activation function, followed by the last fully connected layer with the number of neurons depending on the number of identities in the dataset.

Training and testing process. The training process is done for 100 epochs with a learning rate initialized at 0.1 and reduced by a factor of 0.1 at 40 and 80 epochs. Random erasing [27] is applied in the training process to provide overfitting and improve the classifier's performance. A dropout layer is added before the last layer to provide more regularization. The training process is done using a fine-tuning strategy from CIFAR-10 weights. The testing process is done by extracting features from the first fully connected layer and performing the retrieval process using a ranking algorithm with an additional re-ranking method [28]. The training and testing processes are done for five iterations, and the results are reported by averaging the metric values.

Pre-processing and data augmentation. We applied three pre-processing and data augmentation steps: image resizing, random crop, random horizontal flip, random erasing [27], and data normalization. After loading data into memory, all steps are performed on-fly in a multi-threading fashion.

2.6. Person Re-identification Dataset

The evaluation process is conducted on two widely used person re-identification datasets, Market-1501 and DukeMTMC-reID. This section introduces the dataset and its evaluation protocol before discussing the results.

Market-1501. The dataset is described in [12] and consists of 1,501 unique person identities with a total of 32,668 bounding boxes captured using six cameras. The person bounding box is detected using the DPM people detector [29]. The dataset was divided into training and testing data without any overlapped persons. The training data consists of 751 people with a total number of 12,936 images. Because the retrieval scenario is used in the testing process, the testing data consists of 750 people with 3,368 query images and 19.732 gallery images. The Market-1501 dataset has two different evaluation protocols, single-query, and multi-query. The difference is that the single-query mode is evaluated using only a single image per query. In contrast, the multi-query mode uses multiple images (taken from the same camera) per query. Naturally, the multi-query mode will produce higher accuracy and mAP (mean average precision) than the single-query mode due to more features exposed to the classifier. Fig. 3 shows some examples of the Market-1501 dataset.

Fig. 3. Some person images on Market-1501 person re-identification dataset.

DukeMTMC-reID. The dataset is described in [13] that includes 1,812 people, with 408 people appearing in one camera (treated as gallery distractors) and the rest appearing at least in two cameras. The dataset was split into 702 identities with 16,522 images for training and 19,889 images consisting

of 702 identities for query/gallery set with 408 identities for gallery distractor. Unlike the Market-1501 dataset, the DukeMTMC-reID dataset only consists of the single-query testing evaluation, which is more challenging than the Market-1501 dataset.

3. Results and Discussion

All experiments were done using NVIDIA GPU RTX 2080 TI and PyTorch deep learning framework. To evaluate our proposed classifier, we use practical approaches described in [12] by evaluating our proposed classifier using two different metrics, the mean average precision (mAP) and rank. The metrics compute the distance matrix between the query and the gallery. To ensure that the CNN classifier does not learn the camera characteristics, person images from the same camera were eliminated before retrieval.

The experiments were divided into three scenarios: single classifiers, ensemble classifiers, and ablation study. Experiments using single classifiers are used to evaluate single classifiers of the proposed classifier with several configurations. Experiments using ensemble classifiers are used to evaluate the combination of several single classifiers, which have proven can improve the performance but with an additional number of parameters (due to the combination scheme). The last experiment is the ablation study, which evaluates three factors: the effect of retrieval feature size, re-ranking, and model bias.

3.1. Single Classifier

We summarize the single classifier experiments in Table 1. Those three tables show that the best performance on the Market-1501 dataset is achieved using the first version of the classifier configuration (Res110-CPANv1, Res110-SPANv1, and Res110-SCPANv1). Unlike results on the Market-1501 dataset, the best performance on the DukeMTMC-reID dataset is achieved using the second version of the classifier configuration for SPAN (Res110-SPANv2) and the first version of classifier configuration for other attention types (Res110-CPANv1 and Res110-SCPANv1).

Table 1. Summary of our experiments using single classifier configuration on Market-1501 and DukeMTMC-reID dataset.

#	Classifier	Market-1501						DMTMC-reID		
		Single			*Multi*					
		R@1	*R@5*	*mAP*	*R@1*	*R@5*	*mAP*	*R@1*	*R@5*	*mAP*
LRN classifier with Channel-wise Pyramid Attention Network (CPAN)										
1	Res56-CPANv1	87.84	93.22	82.40	91.66	95.72	86.76	81.63	88.69	75.90
2	Res56-CPANv2	87.14	92.79	81.33	90.81	95.45	85.90	80.22	87.68	74.86
3	Res56-CPANv3	86.92	92.65	80.93	90.72	95.27	85.71	80.69	87.68	75.01
4	Res110-CPANv1	88.97	93.89	83.73	92.32	96.07	87.80	82.24	89.17	76.86
5	Res110-CPANv2	88.22	93.49	82.66	91.81	95.80	86.96	81.71	88.73	76.30
6	Res110-CPANv3	87.85	93.11	82.25	91.61	95.61	86.67	81.73	88.72	76.70
LRN classifier with Spatial Pyramid Attention Network (SPAN)										
1	Res56-SPANv1	88.03	93.40	82.79	91.86	95.82	87.08	81.80	88.70	76.31
2	Res56-SPANv2	87.68	93.17	82.30	91.21	95.52	86.62	81.97	88.78	76.67
3	Res56-SPANv3	87.24	92.77	81.28	90.97	95.31	85.94	80.80	88.15	75.11
4	Res110-SPANv1	88.87	93.88	83.79	92.42	96.06	87.90	82.91	**89.45**	77.53
5	Res110-SPANv2	88.87	93.81	83.70	92.07	95.86	87.67	**83.09**	89.17	**77.88**
6	Res110-SPANv3	87.39	92.93	81.70	91.37	95.64	86.21	80.95	88.52	75.72
LRN classifier with Spatial and Channel-wise Pyramid Attention Network (SCPAN)										
1	Res56-SCPANv1	88.57	93.57	83.12	92.08	95.99	87.46	81.24	88.55	75.92
2	Res56-SCPANv2	87.08	93.03	81.20	91.09	95.49	86.06	81.38	88.11	75.70
3	Res56-SCPANv3	87.21	92.75	81.26	90.78	95.21	85.80	80.56	87.85	75.29
4	Res110-SCPANv1	**89.23**	**93.91**	**83.93**	**92.54**	**96.12**	**88.01**	82.83	89.43	77.79
5	Res110-SCPANv2	88.40	93.45	82.61	91.91	95.75	87.07	82.26	88.99	76.86
6	Res110-SCPANv3	88.19	93.33	82.70	91.62	95.71	87.10	82.15	88.97	76.94

From all different classifier configurations, the Res110-SCPANv1 configuration produces the highest performance on the single-query Market-1501 dataset with the highest rank-1 of 89.23% and mAP of 83.93%. The multi-query evaluation settings on Market-1501 naturally produce higher metrics than single-query. For the multi-query evaluation on the Market-1501 dataset, the highest performance is achieved using the same Res110-SCPANv1 classifier with rank-1 of 92.54% and mAP of 88.01%.

For the DukeMTMC-reID dataset, the proposed classifier achieved the best performance using Res110-SPANv2 with rank-1 of 83.09% and mAP of 77.88%. The SPAN (Spatial Pyramid Attention Network) seems to perform better than the SCPAN attention type on the DukeMTMC-reID dataset. Although the margin between the three types of attention networks is narrow, the results show that the DukeMTMC-reID dataset has more spatial importantness characteristics (some areas are more important than others) than channel feature's importantness.

Fig. 4 shows the heatmap visualization of the last convolutional layer after multiplying by the spatial and channel-wise attention value on the Res110-SCPANv1 classifier. As shown in Fig. 4, the features extracted from the convolutional layer seem very active on a person's upper area (around the head and shoulder) and lower area (around the feet and knee). The results are somehow mimicking Human Visual System (HVS) attention or saliency, which is very sensitive to the face in the images [30]–[32].

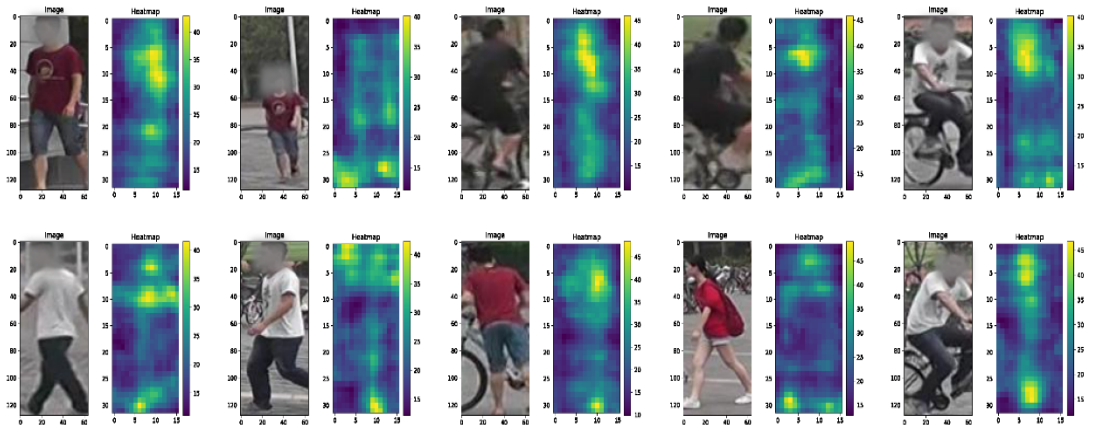

Fig. 4. Heatmap visualization of the last convolutional layer of Res110-SCPANv1 classifier after multiplying by the spatial and channel-wise attention value on Market-1501 dataset (some normalization was applied and the best view on color mode).

The results of the single experiments show that the additional attention and pyramid network can improve the LRN classifier performance, as shown in Table 2. An improvement of ~2% (rank-1 and mAP) is achieved on the Market-1501 dataset, while a ~3% (rank-1 and mAP) improvement is achieved on the DukeMTMC-reID dataset. Although the improvement is impressive, the number of parameters of the classifier is increased by 600K from the original residual network classifier, which is quite significant for the lightweight classifier.

Table 2. Comparison of LRN classifiers with and without attention and pyramid network on Market-1501 and DukeMTMC-reID dataset.

#	Classifier	Market-1501						DMTMC-reID		
		Single			*Multi*					
		R@1	*R@5*	*mAP*	*R@1*	*R@5*	*mAP*	*R@1*	*R@5*	*mAP*
1	Res56	85.93	92.01	80.03	90.13	94.92	84.84	78.93	86.79	72.65
2	Res56-SCPAN (best)	88.57	93.57	83.12	92.08	95.99	87.46	81.24	88.55	75.92
3	Res110	87.12	92.77	81.45	91.17	95.47	86.04	79.98	87.20	74.11
4	Res110-SCPAN (best)	89.23	93.91	83.93	92.54	96.12	88.01	82.83	89.43	77.79

Table 3. Summary of our experiments using Ensemble of several LRN classifiers on Market-1501 and DukeMTMC-reID dataset.

#	Classifier	Market-1501						DMTMC-reID		
		Single			*Multi*					
		R@1	*R@5*	*mAP*	*R@1*	*R@5*	*mAP*	*R@1*	*R@5*	*mAP*
1	Ensemble-1	89.73	94.42	85.25	92.54	96.37	88.94	83.69	89.99	79.32
2	Ensemble-2	90.71	94.77	86.36	93.54	96.61	89.74	84.78	90.78	80.63
3	Ensemble-3	90.70	94.90	86.43	93.43	96.67	89.80	84.76	90.78	80.43
4	Ensemble-4	90.34	94.69	85.99	93.30	96.47	89.50	84.33	90.53	79.98

3.2. Ensemble Classifier

The ensemble approaches were made by concatenated features extracted from each classifier and used for the retrieval process. We use four different ensemble configurations described as follows for the ensemble classifiers experiments.

- **Ensemble-1**. Combination of Res56-CPANv1 and Res56-SCPANv1 features with a total of 1,024 features for the retrieval process.
- **Ensemble-2**. Combination of Res110-CPANv1 and Res110-SCPANv1 features with a total of 1,024 features for the retrieval process.
- **Ensemble-3**. Combination of Res56-SCPANv1 and Res110-SCPANv1 features with a total of 1,024 featuresfor the retrieval process.
- **Ensemble-4**. Combination of Res56-CPANv1, Res56-SCPANv1, Res110-CPANv1, and Res110-SCPANv1 features with a total of 2,048 features for the retrieval process.

Table 3 shows the results of our experiments using four different ensemble classifiers. As shown in Table 3, the ensemble classifier improved the classifier's performance from around 0.5% to 2%. The best performance is achieved using Ensemble-2 with rank-1 of 90.71% and mAP of 86.36% on the single-query Market-1501 dataset; rank-1 of 93.54% and mAP of 89.74% on the multi-query Market-1501 dataset; rank-1 of 84.78% and mAP of 80.63%. Although Ensemble-2 produces the best performance, the performance gap between Ensemble-2 and Ensemble-3 classifiers is very narrow, but Ensemble-3 has fewer parameters than the Ensemble-2 classifier. The last Ensemble-4 classifier consists of more than seven million parameters, but the evaluation shows that the ensemble configuration cannot improve the classifier's performance.

3.3. Comparison

Results of our proposed classifier with other state-of-the-art methods are shown in Table 4. We added additional information density of the classifier (mAP per million parameters and rank-1 per million parameters) and the number of parameters in the classifier for each method to show the classifier's effectiveness. As shown in Table 4, our proposed classifier has not achieved state-of-the-art performance on the Market-1501 or DukeMTMC-reID dataset. The best performance on the Market-1501 dataset is achieved by Wang et al. [33] using a spatial-temporal convolutional neural network. The Viewport-aware convolutional neural network classifier [20] achieved state-of-the-art performance on the DukeMTMC-reID dataset. One of the disadvantages of those two approaches is the classifier has more than 20 million parameters which are considered a large number of parameters in edge-computing or limited-hardware scenarios. The computed information density shows that the state-of-the-art model has low information density, which can be concluded that the model is not effective enough in terms of performance per million parameters. On the other hand, our proposed model has higher information density but lower performance than other state-of-the-art models. The information density of our proposed classifier ranges from 20 to 46, with rank-1 accuracy ranging from 89% to 90%, and mAP ranges from 83.5% to 86%.

Table 4. Comparison of our lightweight classifier with several state-of-the-art models.

Year	Method	#Par. (mil.)	Market-1501						DMTMC-reID		
			Single			*Multi*					
			R@1	*mAP*	*ID*	*R@1*	*mAP*	*ID*	*R@1*	*mAP*	*ID*
2017	CAN [34]	~50	60.3	35.9	1.20 / 0.71	72.1	47.9	1.44 / 0.95	-	-	-
2017	Fisher Net [35]	5.2	48.15	29.94	9.25 / 5.75	-	-	-	-	-	-
2017	Verif-Iden [36]	~23	79.51	59.87	3.45 / 2.60	85.84	70.33	3.73 / 3.05	68.9	49.3	2.99 / 2.14
2018	PAN [37]	~45	88.57	81.53	1.96 / 1.81	91.45	87.44	2.03 / 1.94	75.94	66.74	1.68 / 1.48
2019	Spatial-Temporal [38]	~30	98.0	95.5	3.26 / 3.18	-	-	-	94.0	82.8	3.13 / 2.76
2019	PyrNet [39]	35.9	94.6	91.4	2.63 / 2.54	96.1	94.0	2.67 / 2.61	90.3	87.7	2.51 / 2.44
2019	ARP [40]	~23	87.04	66.89	3.78 / 2.90	-	-	-	73.92	55.56	3.21 / 2.41
2019	HPM [41]	~23	94.20	82.70	4.09 / 3.59	-	-	-	86.60	74.30	3.76 / 3.23
2020	VA-ReID [20]	52.64	96.79	95.43	1.83 / 1.81	-	-	-	93.85	91.82	1.78 / 1.74
2020	SARL [42]	~32	96.1	88.0	3.00 / 2.75	-	-	-	87.9	75.5	2.74 / 2.35
2020	RGA-SC [43]	56.4	96.1	88.4	1.70 / 1.56	-	-	-	-	-	-
2020	Ensemble LRN [26]	2.7	88.61	82.88	32.8 / 30.6	91.77	86.98	33.9 / 32.2	82.49	77.50	30.5 / 28.7
	Res110-SCPANv1 (Our)	2.5	89.23	83.93	35.6 / 33.5	92.54	88.01	37.0 / 35.2	82.83	77.79	33.1 / 31.1
	Res110-CPANv1 (Our)	1.9	88.97	83.73	46.8 / 44.0	92.32	87.80	48.5 / 46.2	82.24	76.86	43.2 / 40.4
	Ensemble-2 (Our)	4.4	90.71	85.25	20.6 / 19.3	93.54	89.74	21.2 / 20.3	84.78	80.63	19.2 / 18.3
	Ensemble-3 (Our)	3.2	90.70	86.43	28.3 / 27.0	93.43	89.80	29.1 / 28.0	84.76	80.43	26.4 / 25.1

3.4. Execution Time

One of the important factors for a lightweight classifier is the execution time to finish the tasks. We use the Market-1501 person re-identification dataset with a single-query scheme to evaluate the execution time of our proposed classifier. The training time, features extraction time, and retrieval process time was calculated as follows. For training time T_{tr}, the time reported is the time elapsed of one epoch, and we only measure using GPU hardware. For features extraction time T_{ex}, retrieval process T_{ret}, and retrieval with the re-ranking process T_{rk}, the time reported is the time elapsed for processing all gallery and query images. Table 5 shows the execution time of our proposed classifier. For comparison, we added the ResNet50-based classifier to the list. We measure the execution time using NVIDIA RTX 2080TI with 11GB memory (GPU hardware) and Intel i7-9700F @3.00 GHz (CPU hardware).

As shown in Table 5, our proposed classifier has a lower execution time compared with the widely used ResNet50-ImageNet classifier, especially for training and feature extraction steps. The execution time for the retrieval process is relatively the same because all of the classifiers in Table 4 use the same number of features for the retrieval process. We use the re-ranking algorithm described in [28], which is not yet optimized for GPU hardware. Although the feature extraction steps' execution time is only half that of the widely used ResNet50-ImageNet classifier, further analysis shows that the time will be higher for a bigger dataset (due to the multiplication factor).

Table 5. Execution time of our proposed classifier for several different processes, the training process, the features extraction process, and the retrieval process.

#	Classifier	System	Market-1501 (in seconds)			
			T_{tr}	T_{ex}	T_{ret}	T_{rk}
1	Res56-CPANv1	CPU	-	193.28 s	14.16 s	52.44 s
		GPU	10.91 s	12.88 s	11.47 s	-
2	Res56-SCPANv1	CPU	-	264.90 s	14.09 s	53.28 s
		GPU	14.43 s	15.00 s	11.41 s	-
3	Res110-CPANv1	CPU	-	294.39 s	14.12 s	53.36 s
		GPU	20.07 s	16.33 s	11.44 s	-
4	Res110-SCPANv1	CPU	-	426.43 s	14.08 s	52.94 s
		GPU	27.14 s	20.55 s	11.44 s	-
5	ResNet50-ImageNet	CPU	-	1025.8 s	14.09 s	52.98 s
		GPU	41.55 s	42.96 s	11.55 s	-

3.5. Ablation Study

In this section, the analysis regarding the effectiveness of our proposed classifier is discussed. We analyze three aspects: the effects of feature size used in the retrieval process, the re-ranking effects, and the bias effects.

1) Retrieval Features Size

One of the factors that may contribute to the performance of a classifier for person re-identification is the size of discriminative features used for the retrieval process. We used three different features to evaluate our proposed classifier, 64, 128, and 256. We use the same setup as used in the main experiments using the Market-1501 dataset with a single-query evaluation and re-ranking algorithm. Fig. 5 shows our proposed classifier's rank-1 and mAP plots using three different fc-sizes. As shown in Fig. 5, the best performance was achieved using 256 discriminant features for the retrieval process. These results are the opposite results of lightweight classifier experiments described in [26], which reported that the best performance is achieved using lower fc-size.

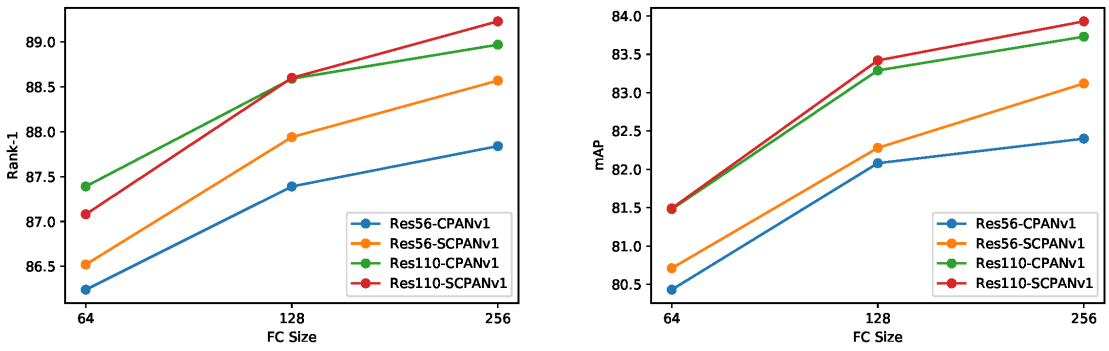

Fig. 5.. Graph of our proposed classifier for different FC sizes, (a) Rank-1, and (b) mAP

2) Re-Ranking Effects

Re-ranking is one way to improve the classifier's performance for person re-identification problems by re-ranking the retrieval results and intuitively using them as query images. The performance comparison between the retrieval process with and without the re-ranking algorithm can be viewed in Fig. 6. As viewed in Fig. 6, the re-ranking algorithm improves the classifier's performance by around 3-4% for rank-1 and 14-17% for mAP. The results show that the re-ranking algorithm increases the classifier confidence, which is indicated by an increase in the mAP value. Although the classifier's performance is increased, the retrieval process time execution was increased around four times when the re-ranking algorithm was used in the process.

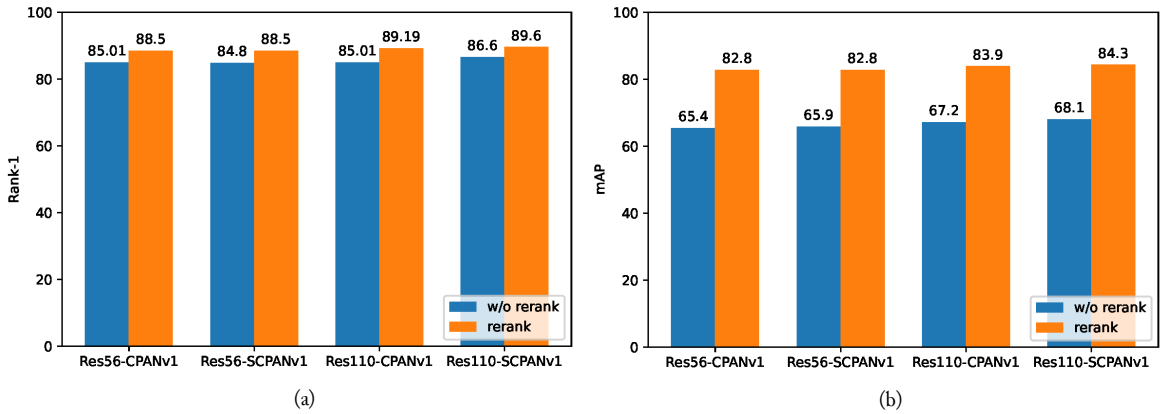

Fig. 6. Graph of our proposed classifier with and without re-ranking algorithm, (a) Rank-1, and (b) mAP.

3) Bias Effects

The last experiment for the ablation study is the bias effects of the classifier by swapping the training and testing data of the Market-1501 and DukeMTMC-reID datasets. The experiments measure the bias effect of the training dataset for general person re-identification problems. Six different SCPAN (Spatial and Channel-wise Pyramid Attention Network) classifiers were tested, and the results can be viewed in Table 6. As shown in Table 6, the proposed classifier trained using the DukeMTMC-reID dataset can achieve rank-1 of around 40% with mAP of around 21% on single-query Market-1501 and rank-1 of around 50% with mAP of around 28% on multi-query Market-1501.

Table 6. The results of bias effects experiments using Market-1501 and DukeMTMC-reID dataset for six different SCPAN classifier.

#	Classifier	D → M						M → D		
		Single			*Multi*			*R@1*	*R@5*	*mAP*
		R@1	*R@5*	*mAP*	*R@1*	*R@5*	*mAP*			
1	Res56-SCPANv1	39.85	53.71	21.48	48.02	61.08	26.27	18.39	27.17	10.71
2	Res56-SCPANv2	44.44	57.34	24.85	52.62	64.07	30.07	19.33	27.76	11.58
3	Res56-SCPANv3	42.21	55.59	23.50	50.48	62.83	28.76	17.78	26.35	10.63
4	Res110-SCPANv1	39.74	53.46	21.58	47.23	60.43	26.33	18.29	27.73	10.99
5	Res110-SCPANv2	43.85	56.77	24.70	51.27	63.37	29.62	19.38	27.63	11.75
6	Res110-SCPANv3	41.12	54.06	22.62	48.82	61.08	27.40	17.80	26.32	10.57

One interesting thing shown in Table 6 is that the Res110 classifier variant has a lower rank-1 and mAP compared with the Res56 variant, although the Res110 consists of more parameters. Unlike the results in the main experiments, the SCPANv2 attention variant has a higher rank-1 and mAP, which means that the attention type has more generality than other attention types. We can also note that our proposed classifier can generalize better when trained using the DukeMTMC-reID dataset compared with the classifier trained using the Market-1501 dataset.

4. Conclusion

We have investigated a lightweight residual classifier with a pyramid attention network for person re-identification. The lightweight classifier was adopted from the residual network originally used for CIFAR experiments which are considered a lightweight classifier due to the number of parameters that not more than two million parameters. Three different pyramid attention networks were used in the experiments, including CPAN (Channel-wise Pyramid Attention Network), SPAN (Spatial Pyramid Attention Network), and SCPAN (Spatial and Channel-wise Pyramid Attention Network). Experiments on Market-1501 and DukeMTMC-reID person re-identification dataset show that with the combination of pyramid attention network, the proposed classifier can achieve rank-1 of more than 92% on Market-1501 and more than 80% on DukeMTMC-reID. Further analysis shows that our proposed

classifier has more information density comparing with other approaches and runs faster on the training and inference process. The proposed lightweight residual classifier with a pyramid attention network is very suitable to deploy on limited computing power scenarios, such as IoT or edge computing. Implementing a complete person re-identification system on limited computing power scenarios can be a topic for future work research. Furthermore, the search for lightweight classifiers is still an open topic, especially for person re-identification and cross-modality person re-identification.

Acknowledgment

This research is partially supported by Indonesia Ministry of Research and Higher Education Research Grant under PDUPT scheme no. 1206/PKS/ITS/2020 and 950/PKS/ITS/2021.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.
Funding statement. This research is partially supported by Indonesia Ministry of Research and Higher Education Research Grant under PDUPT scheme no. 1206/PKS/ITS/2020 and 950/PKS/ITS/2021.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] M. Burchi and V. Vielzeuf, "Efficient Conformer: Progressive Downsampling and Grouped Attention for Automatic Speech Recognition," *2021 IEEE Autom. Speech Recognit. Underst. Work. ASRU 2021 - Proc.*, pp. 8–15, 2021, doi: 10.1109/ASRU51503.2021.9687874.

[2] M. Burchi and R. Timofte, "Audio-Visual Efficient Conformer for Robust Speech Recognition," *Proc. - 2023 IEEE Winter Conf. Appl. Comput. Vision, WACV 2023*, pp. 2257–2266, 2023, doi: 10.1109/WACV56688.2023.00229.

[3] Z. Liu *et al.*, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 9992–10002, 2021, doi: 10.1109/ICCV48922.2021.00986.

[4] X. Huang, M. Dong, J. Li, and X. Guo, "A 3-D-Swin Transformer-Based Hierarchical Contrastive Learning Method for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, doi: 10.1109/TGRS.2022.3202036.

[5] Z. Liu *et al.*, "Video Swin Transformer," Proc. *2022 IEEE/CVF Conf. on Comput. Vision Pattern Recog. (CVPR),* pp. 3192–3201, Sep. 2022, doi: 10.1109/CVPR52688.2022.00320.

[6] Y. C. Wei, Y. X. Lai, and M. E. Wu, "An evaluation of deep learning models for chargeback Fraud detection in online games," *Cluster Comput.*, pp. 1–17, Jul. 2022, doi: 10.1007/S10586-022-03674-4.

[7] Y. Zakaria, M. Hadhoud, and M. Fayek, "Procedural Level Generation for Sokoban via Deep Learning: An Experimental Study," *TechRxiv, Preprint*, Oct. 2021, doi: 10.36227/TECHRXIV.16640095.V3.

[8] H. H. Jebamikyous and R. Kashef, "Autonomous Vehicles Perception (AVP) Using Deep Learning: Modeling, Assessment, and Challenges," *IEEE Access*, vol. 10, pp. 10523–10535, 2022, doi: 10.1109/ACCESS.2022.3144407.

[9] Z. Zhu, Z. Hu, W. Dai, H. Chen, and Z. Lv, "Deep learning for autonomous vehicle and pedestrian interaction safety," *Saf. Sci.*, vol. 145, p. 105479, Jan. 2022, doi: 10.1016/J.SSCI.2021.105479.

[10] Y. Cao, Y. Shao, and H. Zhang, "Study on early warning of E-commerce enterprise financial risk based on deep learning algorithm," *Electron. Commer. Res.*, vol. 22, no. 1, pp. 21–36, Mar. 2022, doi: 10.1007/S10660-020-09454-9.

[11] W. Li, R. Zhao, T. Xiao, and X. Wang, "DeepReID: Deep filter pairing neural network for person re-identification," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 152–159, Sep. 2014, doi: 10.1109/CVPR.2014.27.

[12] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable Person Re-identification: A Benchmark," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1116–

1124. doi: 10.1109/ICCV.2015.133.

[13] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9914 LNCS, pp. 17–35, 2016, doi: 10.1007/978-3-319-48881-3_2.

[14] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled Samples Generated by GAN Improve the Person Re-identification Baseline in Vitro," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-October, pp. 3774–3782, Dec. 2017, doi: 10.1109/ICCV.2017.405.

[15] E. Ustinova, Y. Ganin, and V. Lempitsky, "Multi-Region bilinear convolutional neural networks for person re-identification," *2017 14th IEEE Int. Conf. Adv. Video Signal Based Surveillance, AVSS 2017*, Oct. 2017, doi: 10.1109/AVSS.2017.8078460.

[16] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang, "Beyond Part Models: Person Retrieval with Refined Part Pooling (and A Strong Convolutional Baseline)," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11208 LNCS, pp. 501–518, 2018, doi: 10.1007/978-3-030-01225-0_30.

[17] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, "Joint discriminative and generative learning for person re-identification," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 2133–2142, Jun. 2019, doi: 10.1109/CVPR.2019.00224.

[18] Y. Fu *et al.*, "Self-similarity grouping: A simple unsupervised cross domain adaptation approach for person re-identification," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-October, pp. 6111–6120, Oct. 2019, doi: 10.1109/ICCV.2019.00621.

[19] X. Zhang, J. Cao, C. Shen, and M. You, "Self-training with progressive augmentation for unsupervised cross-domain person re-identification," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-October, pp. 8221–8230, Oct. 2019, doi: 10.1109/ICCV.2019.00831.

[20] Z. Zhu *et al.*, "Viewpoint-Aware Loss with Angular Regularization for Person Re-Identification," *AAAI 2020 - 34th AAAI Conf. Artif. Intell.*, pp. 13114–13121, Dec. 2019, doi: 10.48550/arxiv.1912.01300.

[21] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 3080–3089, Jun. 2019, doi: 10.1109/CVPR.2019.00320.

[22] H. Zhao, J. Jia, and V. Koltun, "Exploring Self-attention for Image Recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 10073–10082, 2020, doi: 10.1109/CVPR42600.2020.01009.

[23] R. F. Rachmadi, K. Uchimura, G. Koutaki, and K. Ogata, "Hierarchical Spatial Pyramid Pooling for Fine-Grained Vehicle Classification," *2018 Int. Work. Big Data Inf. Secur. IWBIS 2018*, pp. 19–24, Sep. 2018, doi: 10.1109/IWBIS.2018.8471695.

[24] D. Zhang, H. Zhang, J. Tang, M. Wang, X. Hua, and Q. Sun, "Feature Pyramid Transformer," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12373 LNCS, pp. 323–339, 2020, doi: 10.1007/978-3-030-58604-1_20.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 770–778, Dec. 2016, doi: 10.1109/CVPR.2016.90.

[26] Y. Dong, H. Liu -, R. Fuad Rachmadi, S. Mardi Susiki Nugroho, and I. Ketut Eddy Purnama, "Lightweight Residual Network for Person Re-identification," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1077, no. 1, p. 012046, Feb. 2021, doi: 10.1088/1757-899X/1077/1/012046.

[27] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random Erasing Data Augmentation," *AAAI 2020 - 34th AAAI Conf. Artif. Intell.*, pp. 13001–13008, Aug. 2017, doi: 10.48550/arxiv.1708.04896.

[28] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 3652–3661, Nov. 2017, doi: 10.1109/CVPR.2017.389.

[29] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp.

1627–1645, 2010, doi: 10.1109/TPAMI.2009.167.

[30] M. Kümmerer, T. S. A. Wallis, and M. Bethge, "Saliency Benchmarking Made Easy: Separating Models, Maps and Metrics," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11220 LNCS, pp. 798–814, 2018, doi: 10.1007/978-3-030-01270-0_47.

[31] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What Do Different Evaluation Metrics Tell Us about Saliency Models?," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 3, pp. 740–757, Mar. 2019, doi: 10.1109/TPAMI.2018.2815601.

[32] T. Judd, F. Durand, and A. Torralba, "A Benchmark of Computational Models of Saliency to Predict Human Fixations," *CSAIL Technical Reports*, Jan. 2012, Accessed: Feb. 05, 2023. Available at: http://hdl.handle.net/1721.1/68590.

[33] G. Wang, J. Lai, P. Huang, and X. Xie, "Spatial-Temporal Person Re-Identification," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 01, pp. 8933–8940, Jul. 2019, doi: 10.1609/AAAI.V33I01.33018933.