

Real-time military person detection and classification system using deep metric learning with electrostatic loss

Suprayitno¹, Willy Achmat Fauzi², Khusnul Ain¹, Moh. Yasin¹

¹Department of Physic, Faculty of Science and Technology, Airlangga University, Surabaya, Indonesia

²Department of Electrical Engineering, Faculty of Industrial Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Article Info

Article history:

Received Jun 19, 2022

Revised Jul 25, 2022

Accepted Sep 30, 2022

Keywords:

Combatants
Convolutional neural network
Coulomb's law
Electrostatic loss
Person detection

ABSTRACT

This study addressed a system design to detect the presence of military personnel (combatants or non-combatants) and civilians in real-time using the convolutional neural network (CNN) and a new loss function called electrostatic loss. The basis of the proposed electrostatic loss is the triplet loss algorithm. Triplet loss' input is a triplet image consisting of an anchor image (x_a), a positive image (x_p), and a negative image (x_n). In triplet loss, x_n will be moved away from x_a but not far from both x_a and x_p . It is possible to create clusters where the intra-class distance becomes large and does not determine the magnitude and direction of x_n displacement. As a result, the convergence condition is more difficult to achieve. Meanwhile, in electrostatic loss, some of these problems are solved by approaching the electrostatic force on charged particles as described in Coulomb's law. With the inception ResNet-v2 128-dimensional vectors network within electrostatic loss, the system was able to produce accuracy values of 0.994681, mean average precision (mAP) of 0.994385, R-precision of 0.992908, adjusted mutual information (AMI) of 0.964917, and normalized mutual information (NMI) of 0.965031.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Khusnul Ain

Department of Physic, Faculty of Science and Technology, Airlangga University

Jl. Airlangga No. 4-6, Airlangga, Surabaya 60115, East Java, Indonesia

Email: k_ain@fst.unair.ac.id

1. INTRODUCTION

Countries globally have prepared themselves for open war. Defense planning has become their focus to deal with conventional warfare [1]. In conventional warfare, especially in conflict areas, it is necessary for soldiers to detect the presence of humans around them and to classify whether the humans detected are combatants, non-combatants, civilians, enemies, or friends. As an anticipatory step for regional security and defense, the determination of human identification and classification can only be done under certain conditions by using the human sense of sight (eyes).

In recent years, digital military applications have increasingly widespread, both in communication and detection [2]–[4]. Researchers have solved the problem of military object detection in various ways. Some of them use hyperspectral imaging and remote sensing [5], distributed sensors [6], [7], and real-time video surveillance [8], [9]. Deep learning on military object detection using video surveillance is also used in weapons installation [10].

Detection and recognition systems are growing rapidly with the discovery of the triplet loss algorithm by Schroff *et al.* [11] which was able to produce an accuracy of 99.63% in the case of facial recognition with the labelled faces in the wild (LFW) dataset. Triplet loss has input in the form of a triplet image consisting of an anchor image (x_a), a positive image (x_p), and a negative image (x_n). The distance between the anchor and the positive will be minimized because they have the same identity while the distance between the anchor and

negative will be maximized because they have different identities [11]. The key to the success of triplet loss is the use of the triplet selection method which is also applied in different applications [12]–[16]. But along with these developments, it turns out that several studies have also emerged as corrections and refinements of triplet loss. A major caveat of the triplet loss, though, is that as the dataset gets larger, the possible number of triplets grows cubically, rendering a long enough training impractical [17], FaceNet picked a random negative for every pair of anchor and positive, which was very time-consuming [18], [19] imports an additional constraint in the traditional triplet loss, which limits the distances of positive pairs to be smaller than a pre-defined value.

This current study analyzed the effectiveness of an application program combined with a camera can identify and classify the presence of humans, thus replacing the function of the eye. This study introduced a new loss function called Electrostatic Loss, modified based on the analysis of the Triplet Loss algorithm associated with electrostatic force, charged particle physics based on Coulomb's Law. Some corrections to Triplet Loss that will be answered in this study are in the discussion of Triplet Loss it is known that (x_n) will be moved away from (x_a) , but does not distance them both $(x_a$ and $x_p)$ [18], does not discuss how close (x_a) and (x_p) is so that it is possible to create clusters where the distance between intra class becomes large [17], and does not determine the magnitude and direction of (x_n) displacement so that convergent conditions will be more difficult to achieve [20]. Therefore, a new Electrostatic Loss algorithm is proposed to give better results.

2. METHOD

2.1. Related work

The loss function is a function that measures the performance of electrostatic loss in predicting the target [21]. The loss function works when the model used makes an error. If the loss function can produce the lowest error, it can be considered to function properly.

2.1.1. History of loss function development

Loss function development as shown in Figure 1 begins with a study on face recognition (FR) using deep FR with deep face [22] and deep ID [23]. After that, the rules of Euclidean distance-based loss known as contrastive loss, triplet loss, and center loss was developed. In 2016 and 2017, L-softmax [24] and A-softmax [25] were designed to promote the development of large-margin feature learning. In 2017, the normalization of features and weights also began to show good performance, which led to research on the softmax variation. In Figure 1, the sections colored red, green, blue, and yellow are representations of a more in-depth softmax method. Where Euclidean distance-based loss, angular/cosine-based loss, and variations of softmax could be identified sequentially. The softmax loss function is a reference generally used in object recognition, but it is not effective enough for FR. The intra-variation can be larger than the inter-difference, and more detailed features are required when recognizing different people [26]. Previous research has robustly focused on creating new loss functions to make features more separable and detailed.

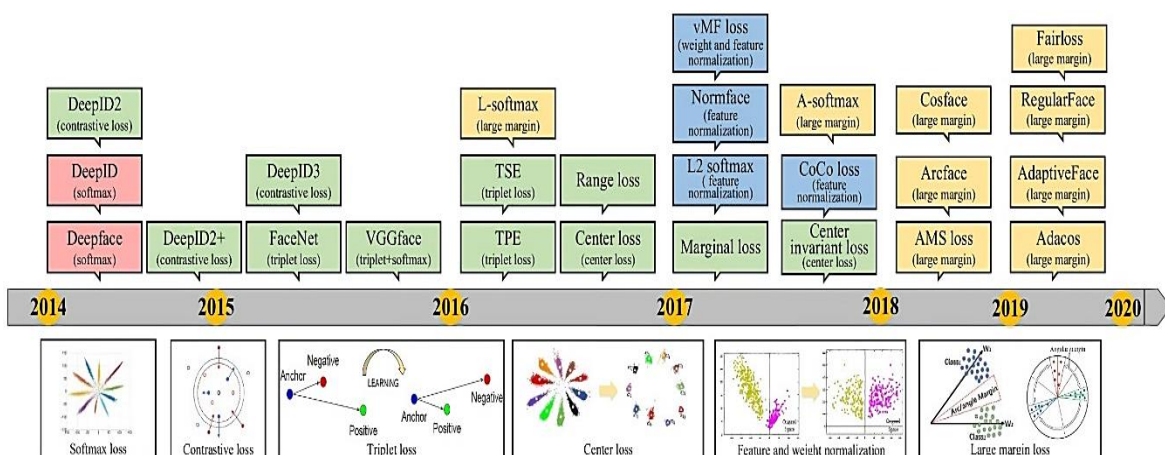


Figure 1. History of loss function development [26]

2.1.2. Triplet loss

Triplet loss is an algorithm that discusses the loss function in the FR process. The triplet loss algorithm is applied in open face as a component of the stochastic gradient descent process during training [27]. It was

first proposed by Schroff *et al.* [11] in the journal of FaceNet, who suggest that FR with the LFW dataset could produce an accuracy of 99.63%. Figure 2 illustrates the mechanism of triplet loss.



Figure 2. Triplet loss minimizes the distance between anchor and positive (both have the same identity information) and maximizes the distance between anchor and negative (both have different identity information) [11]

Triplet loss has input in the form of a triplet image consisting of an anchor image, positive image, and negative image. In a notational form, it can be written as (x_a, x_p, x_n) in which x_a is the anchor image, x_p is the positive image, and x_n is the negative image. Triplet loss aims to minimize the distance between the anchor image and the positive image than with the negative image [11]. Formally, the formula of triplet loss can be seen in (1)-(3):

$$\|x_i^a - x_i^p\|_2^2 + \alpha < \|x_i^a - x_i^n\|_2^2 \quad (1)$$

$$\forall (x_i^a, x_i^p, x_i^n) \in \mathcal{T} \quad (2)$$

$$L = \sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right] \quad (3)$$

where $[Z]_+ = \max(f(z), 0) = \begin{cases} f(z), & \text{if } f(z) > 0 \\ 0, & \text{otherwise} \end{cases}$, and $f(x_i^a); f(x_i^p); f(x_i^n)$ is a feature of the three input images which are generally normalized during training.

In addition, triplet loss has a α parameter, the margin between the positive and negative pair. To measure the similar results of feature extraction between the two images, triplet loss uses Euclidean distance. The parameter in its application is 0.2 to produce a difference between Anchor-positive pair and anchor-negative pair that is relatively far enough and may result in good performance in the training process. Meanwhile, triplet pairs that have a value outside the margin will be ignored as the training process may fail. Thus, it can be concluded that in triplet loss not all available triplet pairs can be used. Meanwhile, \mathcal{T} is all possible triplet sets of images in the training which values are up to N. To summarize, the triplet loss value in (3) can be simplified as (4):

$$L = \max(d(a, p) - d(a, n) + \text{margin}, 0) \quad (4)$$

The symbol a is the anchor image, p is the positive image, and n is the negative image. While d is defined as the distance in the embedding space. From (4), it can be interpreted that triplet loss minimizes Loss by making the value of $d(a, p)$ close to 0 and the value of $d(a, n)$ more than the sum of $d(a, p)$ with a margin.

2.2. Overall framework

This study was designed for end-to-end military detection and recognition based on deep metric learning and the convolutional neural network (CNN), using the concept of loss function named electrostatic loss. The overall framework is divided into two, namely the training and testing process, as well as the production process. Figure 3 shows the training and testing process begins with converting several combatants, non-combatants, and civilian videos into image sequences. The image sequences were then annotated to determine the position of combatants, non-combatants, and civilians. As the annotation results were still of various sizes, the dimensions were homogenized (resizing images) and followed the input according to the architecture. Then, each image was processed with the siamese network, a type of neural network architecture that accepts two or more inputs. These inputs were then processed in the same subnetwork and then combined to calculate the similarity between the two inputs. The output of the siamese network in this study is a vector with N-dimensions. Electrostatic loss as an objective function was then used to determine the error of the siamese network's output. The result of the training is the weight of the network architecture which can later be used for classification, which include combatants, non-combatants, and civilians in this study.

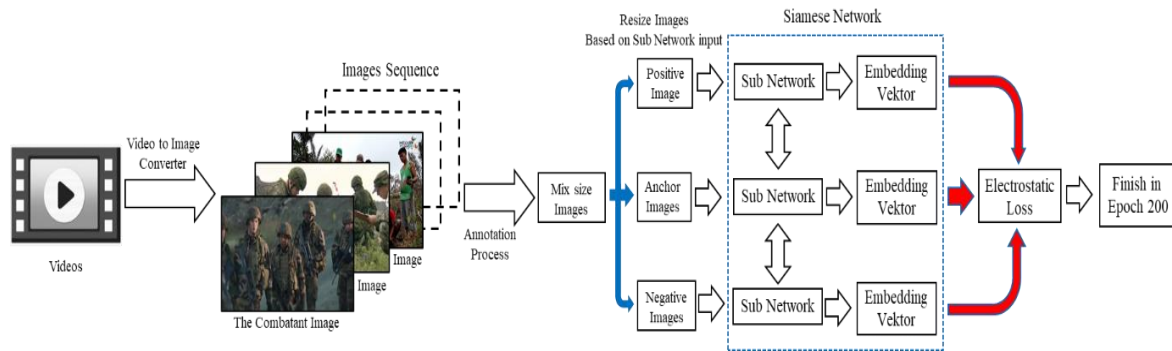


Figure 3. Training and testing process

Figure 4 exhibits that the production process starts from the image produced by the military camera, which would then be processed by MobileNet-SSD as a person detection. The resulting image would then be resized according to the requirements of the input network architecture used. Next, the image would be processed with a network architecture that already weights the previous training process via electrostatic loss. The output of this process was the embedding vector of the image, which would be compared with the center point of the vector image cluster for each class to calculate the distance using the Cosine similarity method. The distance between the vectors shows similarities between images and combatants, non-combatants, or civilians. The closer the image vectors are, the more similar they are to combatants, non-combatants or civilians. If the input image vector is far from the reference vector, it will be classified as unknown.

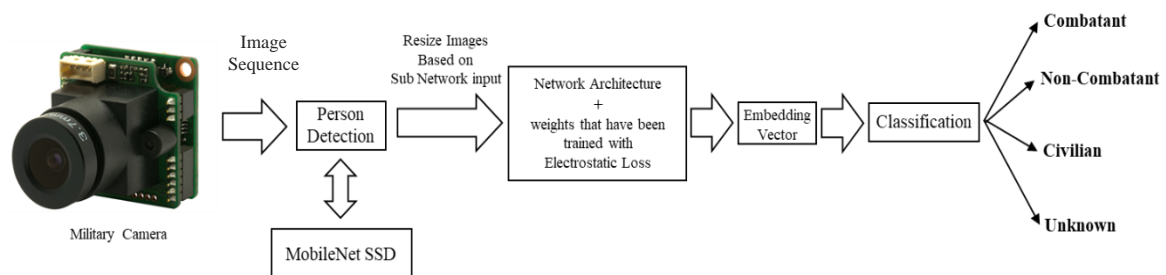


Figure 4. Production process

2.3. Object detection and classification

2.3.1. Object annotation and dataset

To build a system/model design with machine learning, one of the main things needed is a dataset. Due to the scarcity of military dataset, the authors of this study created their own dataset. First, several videos on YouTube that showed the presence of combatants, non-combatants, and civilians were searched. These videos were then converted into image sequences which were annotated to create bounding boxes, positions, and classification for combatants, non-combatants, and civilians. The annotations yielded a dataset in a pascal visual object classes (VOC) format. Figure 5 shows the object annotation process.

2.3.2. Person detection

Figure 4 demonstrates person detection is the initial stage needed as input in the classification process on non-combatants or civilians. In this study, the MobileNet-SSD network architecture was used to detect humans in the prepared video frame. MobileNet is efficiently designed for mobile applications or embedded computer vision [28]. In several studies, the MobileNet-based architecture for object detection reached 4.5 FPS when running on a Raspberry Pi [28].

This detection process shows the position of the bounding box of the person associated with the video frame (x_1, x_2, y_1, y_2) . The objects would be sent to the object classification process. Figure 6 shows a diagram of the MobileNet-SSD network architecture.

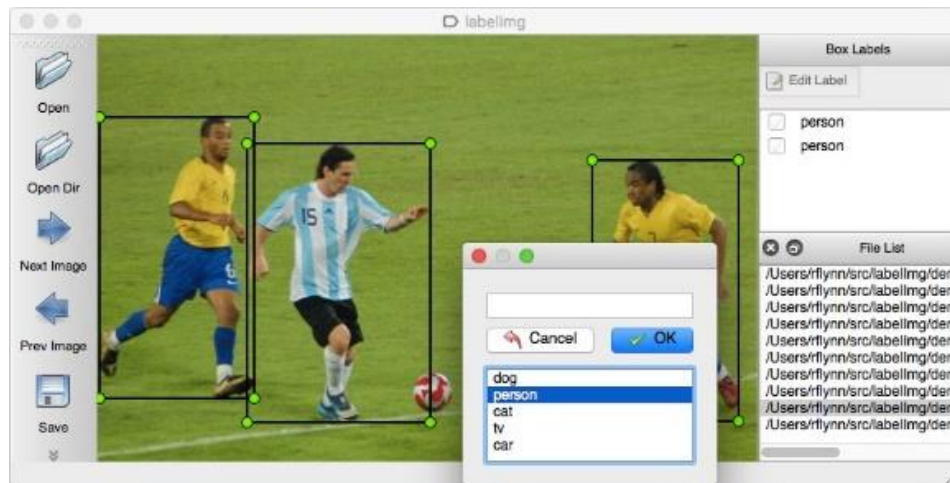


Figure 5. Object annotation process in labelimage [29]

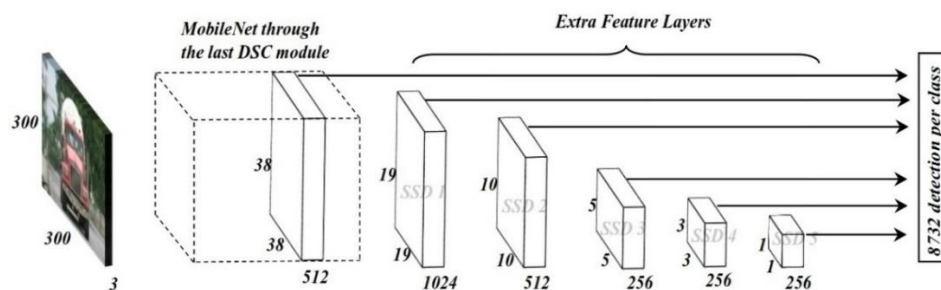


Figure 6. MobileNet-SSD network architecture [30]

2.3.3. Network architecture

In this study, the Inception ResNet-v2 network architecture was used to train and validate the system/model design. Inception was first introduced by Szegedy *et al.* [31] in their research on the development of the CNN in 2014. Recently, very deep convolutional networks have become the main development of image recognition. With its relatively low computation, inception can produce excellent performance.

Several versions of the inception network include Inception-v1 in 2014 [32], Inception-v2, and v3 produced in 2015 [31], as well as Inception-v4, Inception-ResNet-v1, and Inception-ResNet-v2 produced in 2016 [33]. All of these versions developed by Szegedy *et al.* [33] were always renewed from the previous version. In this study, two network architectures, namely Inception-ResNet-v2 and Inception-v4, were compared and used in the next process. The dimensional vector output for Inception-ResNet-v2 and Inception-v4 is 1,000 classes. Therefore, in this study, a fully connected layer was added after the last layer with an N-dimensional size to obtain the N embedding vector, as shown in Figure 7. To learn the Inception-ResNet-v2 design, a new loss function, electrostatic loss, was proposed.

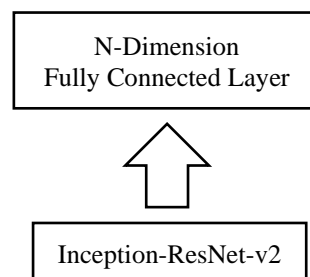


Figure 7. Inception-ResNet-v2 with additional fully connected layer

2.3.4. Embedding dimensionality

In this study, experiments were conducted using several embedding dimensionalities, i.e., 32, 64, and 128, derived from 1,000 classes from two network architectures, namely Inception-Resnet-v2 and Inception-v4. From the embedding dimensionality and network architecture, we could then compare the results of the 32-dimensional Inception-ResNet-v2, 64-dimensional Inception-ResNet-v2, 128-dimensional Inception-ResNet-v2, and 128-dimensional Inception-v4.

2.4. Electrostatic loss

The triplet loss method proposed by Schroff *et al.* [11] becomes the basis of the proposed system. The triplet loss could produce an accuracy of 99.63% in FR with the LFW dataset. It has some algorithm aspects as follows:

- The process of moving $f(x_n)$ away from $f(x_a)$ does not necessarily move $f(x_p)$ away, possibly causing the vector distance between $f(x_a)$ and $f(x_n)$ to not be greater than the vector distance between $f(x_a)$ and $f(x_p)$ [18].
- When an error occurs in choosing an image pair, the error calculation will always be the same and be repeated many times so that it takes longer for the network to converge. Therefore, it applies hard triplet and semi-hard triplet mining [11], [14] for pair selection [17].
- The value of the alpha parameter (α) is a constant value. However, it is quite difficult to determine a good value in a case. Thus, sometimes the selection of the parameter value is based on the researcher's intuition [18], [34].

Based on these findings, electrostatic loss entails a charged particle physics analysis or the electrostatic force. The attractive force and repulsion between charged particles, in this study, are used as an analytical approach in approaching and moving away each vector element from the anchor image ($f(x_a)$), positive image ($f(x_p)$) and negative image ($f(x_n)$). The electrostatic force is stated in (5).

$$F = \left(k \frac{q_1 q_2}{r^2} \right) \quad (5)$$

where:

F = electrostatic force (N)

K = coulomb's constant (Nm^2/C^2)







q_1 = magnitude of charge 1 (C)

q_2 = magnitude of charge 2 (C)

r^2 = distance between charges (m)


According to the Coulomb's law, particles with the same type of charge will experience repulsive force, and particles with different types of charge will experience attractive force. In the application of electrostatic loss, changes are made to the attractive and repulsive force to adjust the conditions that exist in each vector element of the anchor image ($f(x_a)$), positive image ($f(x_p)$), and negative image ($f(x_n)$). In this study, we changed the behavior for the same charge (images) to attract presented in Table 1. However, the basis of force vectors is still based on the Coulomb's law.


Table 1. Adjustment of the rules between Coulomb's law and electrostatic loss to images which are analogous to a charged particle

No.	Charged particle (image)			Coulomb's law	Electrostatic loss
1.		+		Repulsive	Attractive
2.		+		Attractive	Repulsive
3.		+		Attractive	Repulsive

where:

 = anchor (positive)

 = positive

 = negative

Figures 8 and 9 illustrate the analysis of the forces that occur in triplet loss and electrostatic loss. The forces will affect the direction and strength of attraction or repulsion between elements of the vector images. In other words, $f(x_n)$ will move away from $f(x_a)$ and $f(x_p)$. While, $f(x_a)$, $f(x_p)$, and $f(x_n)$ in electrostatic loss are charged particles defined as charge $q \in (0,1)$.

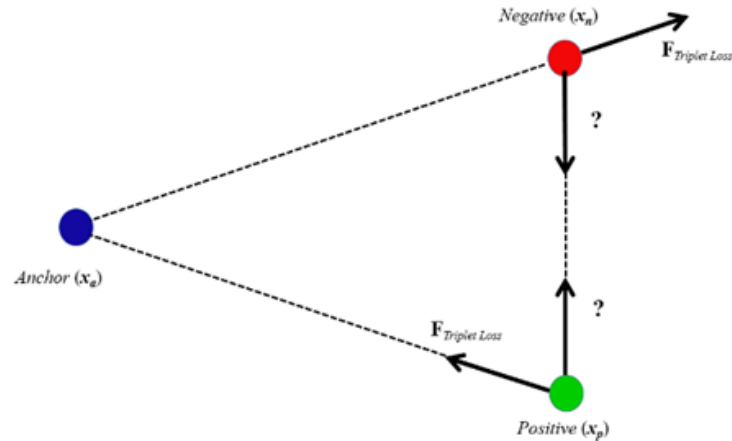


Figure 8. Illustration of style analysis in triplet loss ($F_{\text{triplet loss}}$)

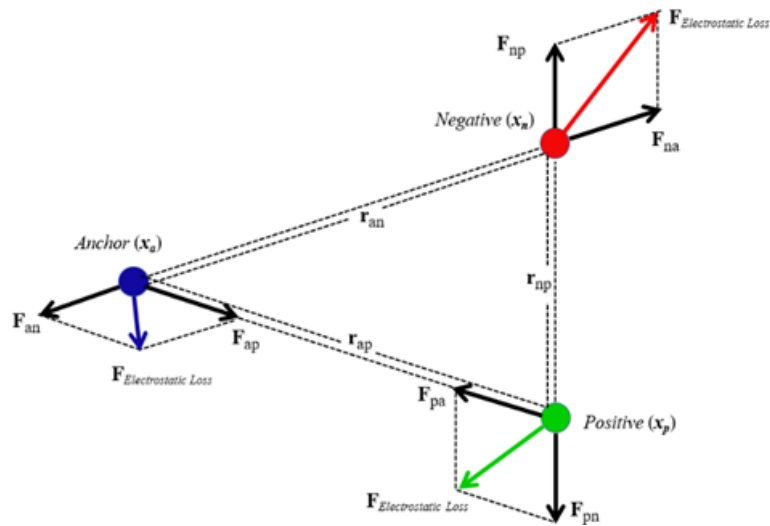


Figure 9. Illustration of force analysis in electrostatic loss ($F_{\text{electrostatic loss}}$)

The value of q is obtained from one of the output values of 128-dimensional vectors. The size of the charge q 's value determines the size of the attractive and repulsive forces between charges. From this concept, a value analysis of electrostatic loss can be calculated as follows (6)-(8):

$$L_{\text{Electrostatic}} = \sum_{a,p,n}^N L_{ap} + L_{an} + L_{pn} \quad (6)$$

$$= \sum_{a,p,n}^N [d(a,p) - d(a,p')]_+ + [d(a,n) - d(a,n')]_+ + [d(p,n) - d(p',n')]_+ \quad (7)$$

$$\begin{aligned} &= \sum_{a,p,n}^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^{p'})\|_2^2 \right]_+ \\ &+ \left[\|f(x_i^a) - f(x_i^n)\|_2^2 - \|f(x_i^a) - f(x_i^{n'})\|_2^2 \right]_+ \\ &+ \left[\|f(x_i^p) - f(x_i^n)\|_2^2 - \|f(x_i^{p'}) - f(x_i^{n'})\|_2^2 \right]_+ \end{aligned} \quad (8)$$

where $[Z]_+ = \max(f(z), 0) = \begin{cases} f(z), & \text{if } f(z) > 0 \\ 0, & \text{otherwise} \end{cases}$ and $f(x_i^a); f(x_i^p); f(x_i^n)$ is a vector value with N dimensions of the anchor, positive and negative images. Meanwhile, $f(x_i^{p'})$ and $f(x_i^{n'})$ are vector values after displacement, influenced by electrostatic forces at a certain time unit. Based on the electrostatic force shown

in (5), it can be explained that the forces experienced by the anchor, positive and negative images can be formulated in (9)-(11).

$$F_{ap} = F_{pa} = \left(k \frac{q_a q_p}{r_{ap}^2} \right) \quad (9)$$

$$F_{an} = F_{na} = \left(k \frac{q_a q_n}{r_{an}^2} \right) \quad (10)$$

$$F_{np} = F_{pn} = \left(k \frac{q_n q_p}{r_{np}^2} \right) \quad (11)$$

q_a , q_p , and q_n are vector value variables taken from the first output embedding index of each anchor, positive and negative image. F_{ap} is the force that arises from vector a to p ; F_{an} is the force that emerges from vector a to n ; and F_{np} is the force that arises from vector n to p .

Determination of other parameter errors can be done by using the velocity variable α which will determine how far the q (charges) should approach (attract) or move away from each other (repulse) in a certain time unit. If the charges approach each other (attract), the position $x_i^{p'}$ can be formulated in (12).

$$x_i^{p'} = F_{pa}\alpha(x_i^p + x_i^a) + F_{pn}\alpha(x_i^p + x_i^n) \quad (12)$$

If the charges move away from each other (repulse), it can be calculated in (13).

$$x_i^{n'} = F_{na}\alpha(x_i^n + x_i^a) + F_{np}\alpha(x_i^n + x_i^p) \quad (13)$$

2.5. Evaluation metrics of research results

The last stage in this study is the evaluation metrics of research result. This stage is carried out to measure and determine the performance of the system/model design that has been made. There are various ways/methods that can be used to measure and determine the performance of a system/model design in research similar to this research. In this study, evaluation metrics of research result was carried out using several methods, namely principal component analysis (PCA), accuracy, mean average precision (mAP), R-precision, adjusted mutual information (AMI), and normalized mutual information (NMI).

2.5.1. Principal componen analysis

The quantity of research data will affect the data analysis. Hence, a technique is needed to simplify/reduce the dimensions of research data. One tool used to reduce the dimensions of data without reducing its characteristics is PCA [35]. The decomposition of the eigenvalues and eigenvectors of the PCA covariance matrix will produce the principal components [36]. The PCA in this study is useful for the visual analysis of embedding vectors at 32, 64, and 128 dimensions.

2.5.2. Accuracy

Accuracy is the percentage of the test set tuples that are correctly classified by the classifier [37]. In other words, the accuracy value is obtained from the comparison between the accurately classified data and the whole data. In the classification process, accuracy measurement is carried out using (14) [37].

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (14)$$

where:

- TP = true positive is the number of positive data that is classified correctly by the system
- TN = true negative is the number of negative data that is classified correctly by the system
- FP = false positive is the number of positive data but classified incorrectly by the system
- FN = false negative is the number of negative data classified incorrectly by the system

2.5.3. Mean average precision

Average precision (AP) and mAP are the most popular metrics used to evaluate an object detection system/model design. AP is calculated individually for each class resulting in AP values as many as the number of classes. These AP scores are averaged to obtain the mAP across all classes. AP is the value obtained from each relevant item precision value which uses a value of 0 for relevant items that are not generated by the

system. The precision value of AP is calculated by taking into account the order of items given by the system. The (15) used to calculate the value of mAP [38].

$$mAP = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision \quad (15)$$

where:

- Q = number of test queries
- R = relevant items generated by the system
- m = the number of relevant items generated from the query

2.5.4. R-precision

R-precision is one of the most frequently used parameters for measuring the accuracy of a system design/recognition model. Although it is empirically stated that R-precision is a good measure, the theoretical reasons are not clear [39]. Aslam and Yilmaz applied a simple geometric interpretation for their research and theoretically proved why R-precision is called a very informative parameter. The value of R-precision on recall r is shown in (16) [39].

$$p(r) = \frac{1-r}{1+\alpha r} \quad (16)$$

Where $\alpha = (1/rp - 1)^2 - 1$. This value of α ensures that the curve passes through the point (rp, rp) .

2.5.5. Adjusted mutual information

The AMI is a variation of mutual information to compare clustering. Specifically, the AMI has a value equal to 1 when the two groupings are identical and 0 when the MI between the two groupings is equal to the expected value [40]. The AMI (17) is defined as (17) [41]:

$$AMI(A, B) = \frac{NMI(A, B) - E\{NMI(A, B)\}}{1 - E\{NMI(A, B)\}} \quad (17)$$

Where $E\{NMI(A, B)\}$ is the expected mutual information between A and B.

2.5.6. Normalized mutual information

The NMI has been widely used as a reference for evaluating metrics on clustering models by measuring the level of dependence between two variables or the level of similarity of information between them [42]. The NMI value itself has a limit from 0 to 1. If the NMI value is 0, then the two variables are independent; if the value is 1, then the variables have the same content. To calculate the NMI, the following (18) can be used [43]:

$$NMI(Y, C) = \frac{2XI(Y; C)}{[H(Y) + H(C)]} \quad (18)$$

where:

- Y = class labels
- C = cluster labels
- H(Y) = entropy value of Y
- H(C) = entropy value of C
- I(Y; C) = mutual information of Y and C

3. RESULT AND DISCUSSION

3.1. Training, testing, and production processes

In this study, the main equipment used was a Core I7-4790 CPU @ 3.60 GHz (8 CPUs) with 32 GB RAM, NVIDIA GeForce GTX 1080 11 GB display, and Python 3.7 programming language. These specifications were used so that the research can run well as expected and adjust to the equipment used by the author in conducting research. The training, testing, and production processes in this study were conducted through preprocessing, feature extraction, and classification.

3.1.1. Preprocessing

The data used in a research process may not always be in ideal conditions nor ready to be processed. Thus, preprocessing is used to improve the quality of the image to be processed for a better analysis.

Preprocessing can reduce the interference that exists in the data. In this study, the preprocessing of the training and testing processes was done in three stages: video conversion to image sequences, object annotation, and image resizing. While the production process went in two stages: person detection and image resizing.

a. Video to image sequence conversion stage

In the training and testing process as well as the production process, some videos obtained from YouTube or military cameras needed to be converted first into image sequences so that they can be processed further. The stages of conversion from video to image sequences were conducted using the video to image converter application. The image sequence in the training and testing processes was used as input for the object annotation stage, while in the production process, the image sequence allowed the MobileNet-SSD to perform the person detection process.

b. Object annotation stage

The object annotation stage was conducted during the training and testing processes by using the label image application. Figures 10-15 show the object annotation results for each class. This step was performed on each image to create a bounding box and was given the position and classification information as combatants, non-combatants, or civilians. From the object annotation process, the research dataset was collected.



Figure 10. One of the images of combatants [44]



Figure 11. The process of annotating to one of combatant images



Figure 12. One of the pictures of non combatants [45]



Figure 13. The process of annotating to one of non combatant images



Figure 14. One of the pictures of civilian [46]



Figure 15. The process of annotating to one of civilian images

From the object annotation stage, the images used as the target dataset were 10,154 combatant images, 10,120 non-combatant images, and 10,118 civilian images, for a total of 30,392 images. The images were then used for training and validating data with as many as 200 epochs. The distribution of images is as follows: i) for the combatant images, the training data consisted of 7,154 images, and the validating data consisted of 3,000 images; ii) for non-combatant images, the training data consisted of 7,120 images, and the validating data consisted of 3,000 images; and iii) for civil images, a total of 7,118 images were used for training data, and 3,000 images were used for validating data.

c. Image resizing stage

The object annotation stage produced combatants, non-combatants, and civilians images that still vary in size. To suit the network architecture used in this study, the images needed to be resized. To follow the requirements of the network architectures in this research (Inception-Resnet-v2 and Inception-v4), the images were resized to 299×299 pixels.

d. Person detection stage

In the training and testing processes, as shown in Figure 3, person detection was performed through Inception-Resnet-v2 or Inception-v4 after the input images were obtained. Inception-Resnet-v2 and Inception-v4 are sub-networks within the siamese network. Meanwhile, in the production process (see Figure 4), person detection was carried out through MobileNet-SSD.

3.1.2. Feature extraction

The images with a size of 299x299 pixels resulting from the preprocessing process became inputs for the feature extraction stage in which the images turn into human features (embedding) in N-dimension. By using the three channels (RGB) image input, the feature extraction step was done to produce N-dimensional vectors. The size of 299×299 pixels was a compatible size with the network architectures used in the next process in the system/model design. Figure 16 is an illustration of N-dimensional vectors of anchor, positive and negative images.

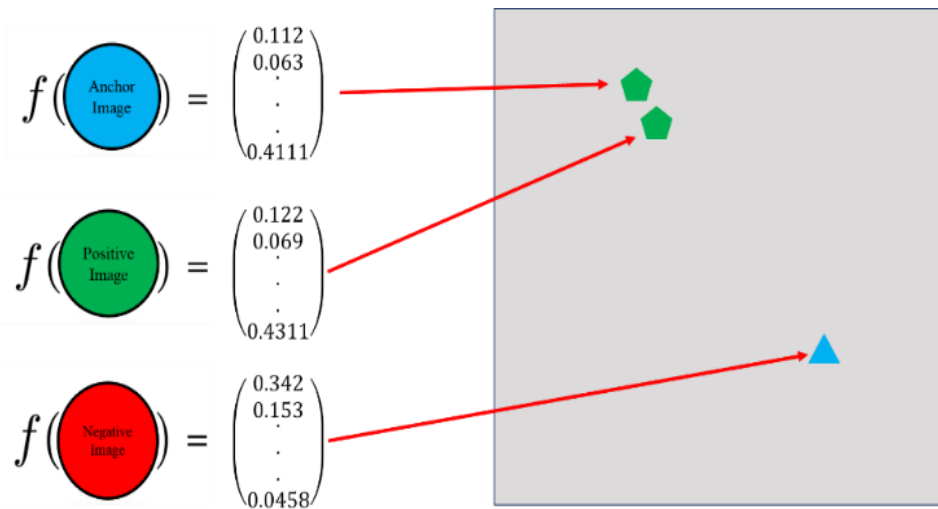


Figure 16. N-dimensional vectors

Electrostatic loss as an objective function in the system/model design serves to minimize the vector distance from the corresponding images and to keep the vector distance from incompatible images, both anchor images, positive images, and negative images. The closer the vector distance is from the corresponding images, the smaller the value of the electrostatic loss is, and vice versa. If the total value of the electrostatic loss decreases and falls close to zero, then the system/model design could be categorized as improving. In addition, if the total value of the electrostatic loss increases, then the system/model design can be categorized as inadequate. The vector distance of these images were calculated using the cosine similarity method, which has been declared better than the euclidean distance method [47].

In this study, a system/model design experiment was conducted using several N-dimensional vectors i.e., 32, 64, and 128. The comparison of the several N-dimensional vectors yielded the result that 128-Dimensional vectors gave the best results. Nevertheless, when the set conditions follow, the maximum error

that is still allowed is below 5% [48]. This result is discussed further in the sub-chapter of evaluation metrics in the next section. Figure 17 shows an example of 128-dimensional vectors.

```
[ -1.8499e-02, 1.5337e-01, -1.5437e-02, 6.3473e-02, 5.3034e-02,
  3.5637e-02, 6.2789e-02, -4.8503e-02, -1.1532e-01, 8.3258e-02,
 -5.1100e-02, 1.3476e-01, 3.9092e-02, 1.2731e-01, 4.1156e-02,
 -4.5459e-03, 4.8917e-02, -4.2416e-02, -3.5232e-03, 7.0245e-02,
 -1.0595e-01, 7.5635e-03, 8.5492e-02, 2.6632e-01, 1.5057e-02,
  2.6204e-02, 6.0506e-02, 8.7383e-02, -4.9148e-02, -1.5678e-01,
  1.2569e-01, -7.7390e-02, 1.0609e-01, -1.3570e-02, -4.8689e-02,
 -1.0025e-01, 3.5710e-02, -3.2984e-02, -4.0355e-02, 2.8508e-01,
  8.1478e-02, -5.6338e-02, -1.7609e-01, -2.6351e-02, 1.6572e-01,
  1.6694e-02, -7.0219e-04, 6.1250e-03, 8.7218e-02, -1.0373e-01,
  1.0257e-01, 1.2167e-01, -5.6445e-02, -1.0016e-01, -5.6210e-03,
  3.5315e-02, -2.4153e-01, 2.2691e-01, 8.1086e-02, 9.0918e-02,
  9.2142e-03, 1.5959e-01, 1.1637e-01, 5.1782e-02, -1.0491e-01,
 -1.3851e-01, 1.1413e-01, 1.0464e-01, 3.1668e-02, -2.2503e-02,
  1.6693e-01, 1.4645e-01, 2.3553e-01, 1.9978e-02, -4.3823e-03,
  4.6144e-03, -1.5295e-02, 5.6460e-02, 1.4087e-01, 5.4563e-02,
 -7.5930e-02, 7.0777e-02, 2.5873e-02, -9.8215e-02, -1.1899e-01,
  1.3139e-01, -6.5762e-02, 2.9281e-01, 3.4578e-01, 7.9961e-02,
  1.4072e-02, 1.6696e-03, 5.6733e-02, -5.8436e-03, -5.5403e-03,
  9.2028e-02, 6.9629e-02, 1.2371e-01, -1.0444e-01, 8.5556e-02,
 -2.5981e-02, -1.0778e-01, -5.4768e-02, 4.4301e-02, -2.7666e-01,
 -1.0240e-01, 3.8106e-02, 9.4536e-02, -1.4280e-01, -2.2915e-03,
 -1.3292e-01, 2.9621e-02, 6.9914e-02, -1.9917e-01, 8.2756e-02,
 -1.5844e-01, -3.1255e-02, -4.5193e-02, -9.1559e-02, -1.3888e-02,
  4.8800e-02, 1.7091e-01, 1.7088e-01, -7.4453e-02, 8.7954e-02,
  7.2497e-02, 1.5243e-01, -1.3949e-01]
```

Figure 17. 128-dimensional vectors

3.1.3. Classification

Classification is a data analysis process that produces models to describe classes contained in data [37]. A system/model design can be created from the classification of training data provided and used on new data. It is expected that the system is able to classify all data correctly. Therefore, some adjustments are necessary to make the error smaller and closer to zero. In this study, after the human features (embedding) were obtained, they were then identified and classified by three classes, namely combatants, non-combatants, or civilians. When it does not match the three classes it will be categorized as unknown.

3.2. Evaluation metrics analysis of research results

As explained in the previous chapter, the methods used to evaluate the system design/model include the PCA, accuracy, mAP, R-precision, AMI, and NMI method. The training process and system/model validation were conducted for up to 200 epochs under several conditions. The first step of the experiment is changing dimensional vectors' sizes into 32, 64, and 128 dimensions followed by changing the network architecture into Inception ResNet-v2 and Inception-v4. In the last step, changing loss function is done using triplet loss and electrostatic loss.

3.2.1. Principal component analysis

The purpose of PCA on a system/model design is to facilitate observations. With PCA, dimensions are reduced from 128-dimensional vectors to 3-dimensional vectors. Figures 18-21 demonstrate dimensional vectors using PCA. Figure 18 exhibits the result of PCA on Inception ResNet-v2 32-Dimensional vectors. Figure 19 shows the result of PCA on Inception ResNet-v2 64-dimensional vectors. Figure 20 explains the PCA results on Inception ResNet-v2 128-dimensional vectors. Lastly, Figure 21 indicates the PCA results on Inception-v4 128-dimensional vectors. Among the PCA results in different dimensional vectors, the Inception ResNet-v2 128-dimensional vectors network architecture had the best result because it had the smallest distance between vectors within the same class (interclass) and the largest distance between vectors in different classes (intra-class).

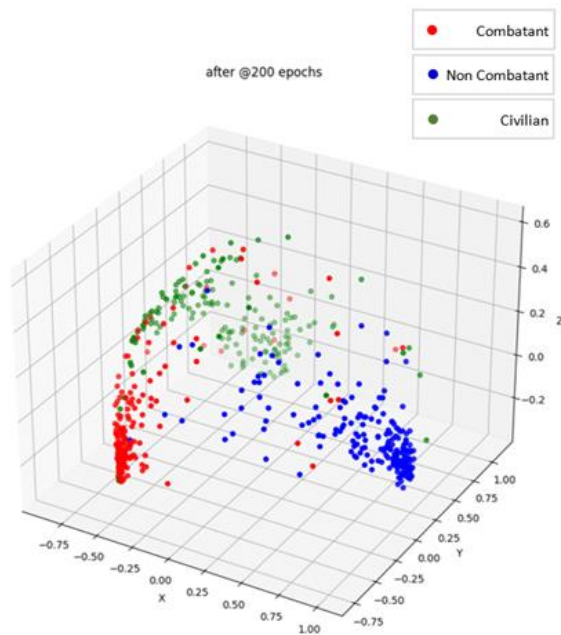


Figure 18. PCA results on Inception ResNet-v2 32-dimensional vectors into 3-dimensional vectors

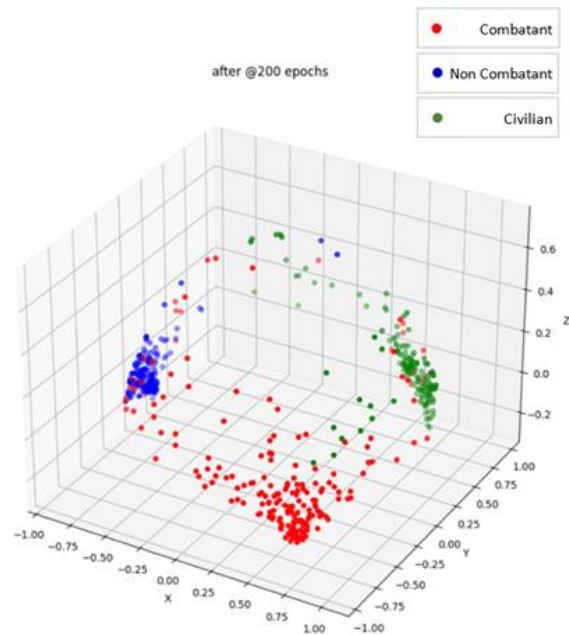


Figure 19. PCA results on Inception ResNet-v2 64-dimensional vectors into 3-dimensional vectors

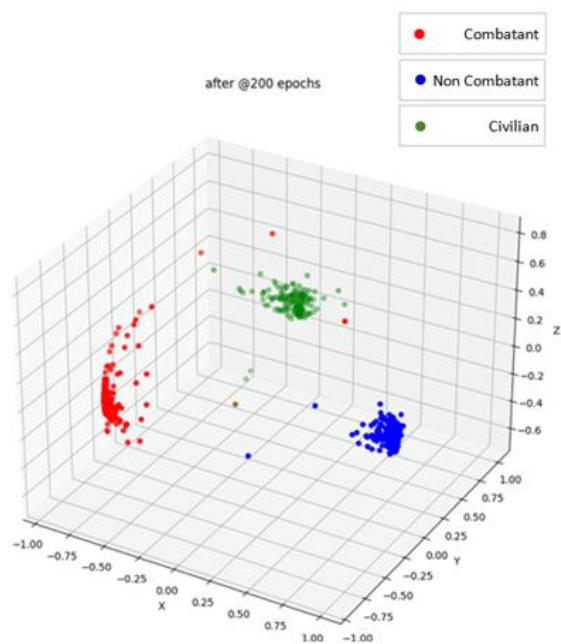


Figure 20. Results on Inception ResNet-v2 128-dimensional vectors into 3-dimensional vectors

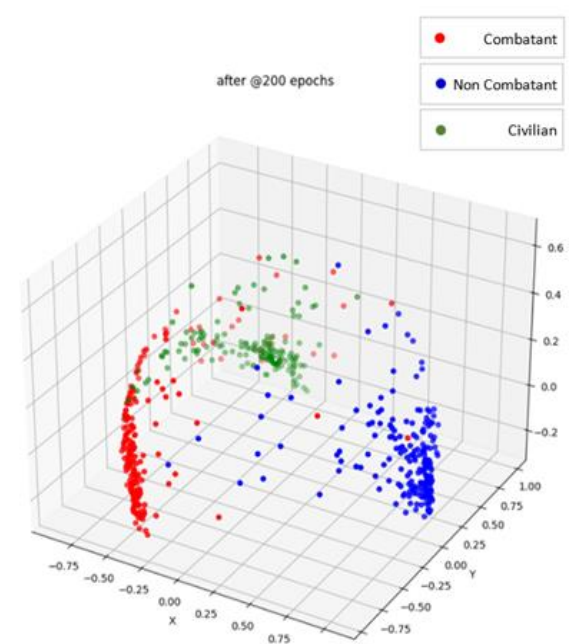


Figure 21. PCA results on Inception-v4 128-dimensional vectors into 3-dimensional vectors

3.2.2. Accuracy, mean average precision, R-precision, adjusted mutual information, and normalized mutual information analyses

The use of accuracy, mAP, and R-precision for a system/model design analysis aims to measure the accuracy level of the classification in the system. While AMI and NMI are used to measure the quality of the class clusters from the system design/research model, as well as the distance between the interclass and intraclass vectors. This study included three analyses to find the differences in dimensional vectors, in network architecture, and in loss functions, respectively.

a. Analysis of different dimensional vectors

With different dimensional vectors (32, 64, and 128) and a fixed network architecture, Inception Resnet-v2, the system/model design trial process showed different accuracy values (see Table 2). Table 2 indicates that the highest value was generated by the 128-dimensional-vector system (accuracy values =0.994681, mAP=0.994385, R-precision=0.992908, AMI=0.964917, and NMI=0.965031). Results concluded the larger the dimensional vectors, the greater the iteration time, and the better the results as also seen in previous research conducted by Schroff *et al.* [11].

Tabel 2. The values of accuracy resulted from mAP and R-precision, AMI and NMI Analysis using electrostatic loss with different dimensional vectors 32, 64, and 128

Name	Accuracy	mAP	R-precision	AMI	NMI
Electrostatic loss [32] Inception ResNet-v2	0.875887	0.883422	0.880024	0.650689	0.651830
Electrostatic loss [64] Inception ResNet-v2	0.907801	0.915632	0.906028	0.686539	0.687562
Electrostatic loss [128] Inception ResNet-v2	0.994681	0.994385	0.992908	0.964917	0.965031

b. Analysis of different network architectures

In the other experiment, the Inception ResNet-v2 and Inception-v4 network architectures were used with electrostatic loss. Inception ResNet-v2 is the latest network architecture developed from the previous version of the network architecture by Szegedy *et al.* [33]. Besides, Inception ResNet-v2 and Inception-v4 have an accuracy of more than 80% and an operating cost of fewer than 15 G-FLOPs. Although several network architectures have an accuracy of more than 80% (NASNET-A-large and SENet 154), they are not used due to their operating cost of more than 20 G-FLOPs and memory usage greater than Inception ResNet-v2 and Inception-v4 [49]. Table 3 depicts the results of the system/model design trial process with the aforementioned settings. Table 3 indicates that the Inception ResNet-v2 network architecture system generates the highest value (accuracy values=0.994681, mAP=0.994385, R-precision=0.992908, AMI=0.964917, and NMI=0.965031).

Tabel 3. The values of accuracy from mAP, R-precision, AMI, and NMI analyses of the system/model designs using electrostatic loss and Inception ResNet-v2 and Inception-v4

Name	Accuracy	mAP	R-precision	AMI	NMI
Electrostatic loss [128] Inception-v4	0.898936	0.910757	0.908983	0.756541	0.757332
Electrostatic loss [128] Inception ResNet-v2	0.994681	0.994385	0.992908	0.964917	0.965031

c. Analysis of different loss function

In designing a system/model, this study employed different loss functions, namely triplet loss and electrostatic loss. Table 4 shows the results of system/model design trial process using different loss functions. Table 4 indicates that the system/model design trained using electrostatic loss produced a higher value than triplet loss (accuracy value=0.994681, mAP=0.994385, R-precision=0.992908, AMI=0.964917, and NMI=0.965031). Electrostatic loss is developed from triplet loss in which a calculation term was added to the distance between positive and negative images. In addition, Coulomb's law was also added to reduce the inter-class distance and increase the intraclass distance according to (5).

Tabel 4. The accuracy values from mAP, and R-precision analyses of the system/model designs within the [128] Inception ResNet-v2 network architecture with triplet loss and electrostatic loss

Name	Accuracy	mAP	R-precision	AMI	NMI
Triplet loss [128] Inception ResNet-v2	0.918440	0.921099	0.916667	0.738010	0.738868
Electrostatic loss [128] Inception ResNet-v2	0.994681	0.994385	0.992908	0.964917	0.965031

4. CONCLUSION

Based on the results analyses, this study concludes. Regarding the ability of detection and classification, electrostatic loss produces better accuracy than triplet loss. The use of electrostatic force based on the Coulomb's law suggests removing $f(x_n)$ from $f(x_a)$ and $f(x_p)$ is taken to prevent the vector distance between $f(x_a)$ and $f(x_n)$ smaller than that between $f(x_a)$ and $f(x_p)$. It is necessary to adjust the rules between Coulomb's law and Electrostatic Loss to images which are analogous to a charged particle. With electrostatic loss, Inception ResNet-v2 with 128-dimensional vectors yields the best network architecture. Electrostatic loss can be used as an alternative loss function through the deep metric learning.

The results of this study still warrant further investigation. Future research may conduct the following actions. Using electrostatic loss in designing a system/model with more classes, datasets, and dimensional vectors higher than 128. Using other equipment that has a higher capability than the equipment used in this study

ACKNOWLEDGEMENTS

The authors greatly acknowledge the support from Department of Physic, Faculty of Science and Technology, Airlangga University, Surabaya, Indonesia, for providing the necessary resources to carry out this research work and the reviewers and journal editorial board for their many insightful comments, which have significantly improved this article. The authors also express their deepest gratitude to the Ministry of Education, Culture, Research and Technology of the Republic of Indonesia for the financial support for this research through the Simlitabmas program.




REFERENCES

- [1] J. M. Taw, "Planning for military operations other than war : Lessons from US army efforts," *Australian Defence Force Journal*, vol. 134, pp. 57–66, 1999.
- [2] J. J. Kang, W. Yang, G. Dermody, M. Ghasemian, S. Adibi, and P. Haskell-Dowland, "No soldiers left behind: An IoT-based low-power military mobile health system design," *IEEE Access*, vol. 8, pp. 201498–201515, 2020, doi: 10.1109/ACCESS.2020.3035812.
- [3] K. S. Chan and F. T. Johnsen, "Military communications," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 10–10, Oct. 2017, doi: 10.1109/MCOM.2017.8067675.
- [4] V. Gotarane and S. Raskar, "IoT practices in military applications," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Apr. 2019, pp. 891–894, doi: 10.1109/ICOEI.2019.8862559.
- [5] C. Ke, "Military object detection using multiple information extracted from hyperspectral imagery," in *2017 International Conference on Progress in Informatics and Computing (PIC)*, Dec. 2017, pp. 124–128, doi: 10.1109/PIC.2017.8359527.
- [6] S. Astapov, J.-S. Preden, J. Ehala, and A. Riid, "Object detection for military surveillance using distributed multimodal smart sensors," in *2014 19th International Conference on Digital Signal Processing*, Aug. 2014, pp. 366–371, doi: 10.1109/ICDSP.2014.6900688.
- [7] C. V. Mahamuni, "A military surveillance system based on wireless sensor networks with extended coverage life," in *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, Dec. 2016, pp. 375–381, doi: 10.1109/ICGTSPICC.2016.7955331.
- [8] R. Benzer and M. C. Yildiz, "YOLO approach in digital object definition in military systems," in *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, Dec. 2018, pp. 35–37, doi: 10.1109/IBIGDELFT.2018.8625314.
- [9] W. Zhang, J. Li, and S. Qi, "Object detection in aerial images based on cascaded CNN," in *2018 International Conference on Sensor Networks and Signal Processing (SNSP)*, Oct. 2018, pp. 434–439, doi: 10.1109/SNSP.2018.00088.
- [10] R. Bisewski and P. K. Atrey, "Toward a remote-controlled weapon-equipped camera surveillance system," in *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, Nov. 2011, pp. 1087–1092, doi: 10.1109/ICTAI.2011.185.
- [11] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 815–823, doi: 10.1109/CVPR.2015.7298682.
- [12] Y. Cui, F. Zhou, Y. Lin, and S. Belongie, "Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 1153–1162, doi: 10.1109/CVPR.2016.130.
- [13] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International Workshop on Similarity-Based Pattern Recognition*, 2015, pp. 84–92, doi: 10.1007/978-3-319-24261-3_7.
- [14] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proceedings of the British Machine Vision Conference 2015*, 2015, pp. 1–12, doi: 10.5244/C.29.41.
- [15] J. Wang *et al.*, "Learning fine-grained image similarity with deep ranking," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 1386–1393, doi: 10.1109/CVPR.2014.180.
- [16] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 5005–5013, doi: 10.1109/CVPR.2016.541.
- [17] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, arxiv.org/abs/1703.07737.
- [18] Y. Yuan, W. Chen, Y. Yang, and Z. Wang, "In defense of the triplet loss again: Learning robust person re-identification with fast approximated triplet loss and label distillation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2020, pp. 1454–1463, doi: 10.1109/CVPRW50498.2020.00185.
- [19] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based CNN with improved triplet loss function," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 1335–1344, doi: 10.1109/CVPR.2016.149.
- [20] H. Xuan and R. Pless, "Extreme triplet learning: Effectively optimizing easy positives and hard negatives," *ICLR 2020 open review*, vol. 53, no. 3, pp. 662–675, 2020.
- [21] X. Zhu, H.-I. Suk, and D. Shen, "A novel matrix-similarity based loss function for joint regression and classification in AD diagnosis," *NeuroImage*, vol. 100, pp. 91–105, Oct. 2014, doi: 10.1016/j.neuroimage.2014.05.078.
- [22] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 1701–1708, doi: 10.1109/CVPR.2014.220.
- [23] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 1891–1898, doi: 10.1109/CVPR.2014.244.
- [24] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," 2016, arxiv.org/abs/1612.02295




- [25] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 6738–6746, doi: 10.1109/CVPR.2017.713.
- [26] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, Mar. 2021, doi: 10.1016/j.neucom.2020.10.081.
- [27] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "OpenFace: A general-purpose face recognition library with mobile applications," *CMU School of Computer Science*, vol. 6, no. 2, pp. 1–18, 2016, [Online]. Available: <http://cmusatyalab.github.io/openface/>
- [28] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017, arxiv.org/abs/1704.04861.
- [29] Tzatalin, "LabelImg," *Github*. [Online]. Available: <https://github.com/heartexlabs/labelImg>, [May 3, 2017]
- [30] S. Arabi, A. Haghighat, and A. Sharma, "A deep learning based solution for construction equipment detection: From development to deployment," 2019, arxiv.org/abs/1904.09021.
- [31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.
- [32] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.
- [33] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Feb. 2017, pp. 4278–4284, doi: 10.1609/aaai.v31i1.11231.
- [34] S. Biswas and J. Gall, "Multiple instance triplet loss for weakly supervised multi-label action localisation of interacting persons," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Oct. 2021, pp. 2159–2167, doi: 10.1109/ICCVW54120.2021.00245.
- [35] Adiwijaya, U. N. Wisesty, E. Lisnawati, A. Aditsania, and D. S. Kusumo, "Dimensionality reduction using Principal Component Analysis for cancer detection based on microarray data classification," *Journal of Computer Science*, vol. 14, no. 11, pp. 1521–1530, Nov. 2018, doi: 10.3844/jcssp.2018.1521.1530.
- [36] A. Jamal, A. Handayani, A. A. Septiandri, E. Ripmiation, and Y. Effendi, "Dimensionality reduction using PCA and K-means clustering for breast cancer prediction," *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, vol. 9, no. 3, pp. 192–201, Dec. 2018, doi: 10.24843/LKJITI.2018.v09.i03.p08.
- [37] J. Han, M. Kamber, and J. Pei, *Data mining concepts and technique*. Amsterdam: Elsevier, 2012, doi: 10.1016/C2009-0-61819-5.
- [38] C. D. Manning, P. Raghavan, and H. Schütze, *An introduction to information retrieval*. Cambridge: Cambridge University Press, 2008.
- [39] J. A. Aslam and E. Yilmaz, "A geometric interpretation and analysis of R-precision," in *Proceedings of the 14th ACM international conference on Information and knowledge management - CIKM '05*, 2005, pp. 664–671, doi: 10.1145/1099554.1099721.
- [40] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.
- [41] A. Amelio and C. Pizzuti, "Correction for closeness: Adjusting normalized mutual information measure for clustering comparison," *Computational Intelligence*, vol. 33, no. 3, pp. 579–601, Aug. 2017, doi: 10.1111/coin.12100.
- [42] P. Zhang, "Evaluating accuracy of community detection using the relative normalized mutual information," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2015, no. 11, pp. 1–9, Nov. 2015, doi: 10.1088/1742-5468/2015/11/P11006.
- [43] T. O. Kvalseth, "Entropy and correlation: Some comments," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 17, no. 3, pp. 517–519, 1987, doi: 10.1109/TSMC.1987.4309069.
- [44] Military TV. *Top 20 military uniform patterns / top 20 military clothing camouflage patterns*. [Online]. Available: <https://www.youtube.com/watch?v=ITK39C6wBGg>, [Feb. 14, 2020].
- [45] Truth Duty Valour. *Truth duty valour episode 304 – combat medics*. (Mar. 17, 2009). Accessed: [Online]. Available: <https://www.youtube.com/watch?v=VlrwJ9PK35k>, [Mar. 17, 2009]
- [46] Medan Today, Martelu, Deli Serdang, Sumatera utara, Indonesia. *Martelu village communities take care and protect communal land [Indonesian]*. [Online]. Available: <https://www.youtube.com/watch?v=oRsYJaLMFkM>, [Nov. 29, 2020].
- [47] O. E. Oduntan, I. A. Adeyanju, A. S. Falohun, and O. O. Obe, "A comparative analysis of euclidean distance and cosine similarity measure for automated essay-type grading," *Journal of Engineering and Applied Sciences*, vol. 13, no. 11, pp. 4198–4204, 2018, doi: 10.36478/jeasci.2018.4198.4204.
- [48] E. Ruel, "What is a survey?," in *100 Questions (and Answers) About Survey Research*, 2455 Teller Road, Thousand Oaks California 91320: SAGE Publications, Inc., 2019, pp. 3–3, doi: 10.4135/9781506348803.n5.
- [49] S. Bianco, R. Cadene, L. Celona, and P. Napolitano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64270–64277, 2018, doi: 10.1109/ACCESS.2018.2877890.

BIOGRAPHIES OF AUTHORS






Suprayitno    received his bachelor's degree in 1997 at Department of Physics, Faculty of Science and Technology, Airlangga University, and master degrees in 2011 from Information Technology Management, Sepuluh Nopember Institute of Technology. All of them are in Surabaya, Indonesia. His current research interests include microcontrollers, image processing, and the internet of things. He can be contacted at email: suprayitno-2017@fst.unair.ac.id or prayit.suprayitno.sp@gmail.com.






Willy Achmat Fauzi    studied Diploma 4 at Politeknik Elektronika Negeri Surabaya (PENS) from 2005 to 2009. His master degree at the Department of Electrical Engineering Institut Teknologi Sepuluh Nopember Surabaya, Indonesia in Multimedia Smart Networks since 2018 until 2021. His current research interests include image processing, deep metric learning, and the internet of things. He can be contacted at email: willy.achmat@gmail.com.



Khusnul Ain    is a lecturer in Bachelor of Biomedical Engineering, Department of Physics, Faculty of Science and Technology, Airlangga University, Indonesia. He got his bachelor, master, and doctoral degree from Gadjah Mada University majoring Nuclear Engineering, Physics, and Bandung Institute of Technology majoring Physics Engineering. All of them are in Indonesia. His research field are tomography and spectroscopy of electrical bioimpedance. He can be contacted at email: k_ain@fst.unair.ac.id.



Moh. Yasin    received his B.Sc. in 1990 at Airlangga University, and M.Sc. and Ph.D degrees from Gadjah Mada University in 1999 and 2010, respectively. Currently, he is an associate professor at the Faculty of Science and Technology, Airlangga University, Indonesia. His research interests include fiber optic communications and sensors. He can be contacted at email: yasin@fst.unair.ac.id.