

A Convolutional Neural Network with K-Nearest Neighbor for Image Classification

Fatima Ahmed Abubakar¹, Souley Boukari²

MSc Student, Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi, Nigeria¹

Professor, Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi, Nigeria²

Abstract: Image classification forms the basis for computer vision which is a trending sub-field in Machine Learning. The Convolutional Neural Network (ConvNet) has recently achieved great success in many computer vision tasks. The common architecture of ConvNets contains many layers to recurrently extract suitable image features and feed them to the softmax function for classification which often displays low prediction performance. In this paper, we propose the use of K-Nearest Neighbor as classifier for the ConvNets and also introduce the use of Principal Component Analysis (PCA) for dimensionality reduction. When successfully implemented, the proposed system should be able to accurately classify images.

Keywords: Image classification, Convolutional Neural Networks, Principal Component Analysis, K-Nearest Neighbor.

I. INTRODUCTION

The Convolutional Neural Network (ConvNet) has recently achieved great success in many computer vision tasks. ConvNet was partially inspired by neuroscience and thus shares many of the brain's properties. Training a ConvNet can be achieved by back-propagating the classification error, which requires a reasonable amount of training data based on the size of the network.

Image classification can be defined as the task of categorizing images into one of several predefined classes, is a fundamental problem in computer vision. According to [1], image classification forms the basis for other computer vision tasks such as localization, detection and segmentation. Image classification is an important task in the field of machine learning and image processing, which is widely used in many fields, such as computer vision, network image retrieval and military automation target identification.

ConvNets, as standard feature extractors, have been continuously used to improve computer vision in terms of accuracy. This implies expulsion of the traditional hand-crafted feature extraction techniques in computer vision problems. The features learned from ConvNets are generated using a general-purpose learning procedure [2]. Combining both hand-crafted features and machine learned feature is increasingly becoming a hot spot [3].

The common architecture of ConvNets contains many layers to recurrently extract suitable image features and feed them to the softmax function (also known as multinomial logistic regression) for classification [4]. [5] and [6] replaced softmax with Biomimetic Pattern Recognition (BPR) and Support Vector Machine (SVM) respectively in order to overcome the limitation of softmax classifier which according to [7], often displays a low prediction performance.

According to [8], at every stage of unsupervised learning K-Nearest Neighbor(K-NN) can perform better than the SVM. Principal Component Analysis can also be used to reduce the dimension of the convoluted image [9].

In view of the above problem stated, this work is aimed at proposing an improved Convolutional Neural Network (ConvNet) with reduced complexity and improve precision that can be easily trained and can adapt to different data and tasks for image classification.

To achieve this, K-Nearest Neighbor (KNN) is going to be adopted as the classifier while PCA for image dimension reduction will be adopted at the fully connected layer before input into the K-Nearest Neighbor (KNN) for classification as this to reduce the complexity.

Principal Component Analysis (PCA) is one of the statistical techniques frequently used in signal processing, data dimension reduction or data decorrelation. PCA is often used in signal and image processing as it offers a powerful means for data analysis and pattern recognition which are used as a technique for data compression, dimension

reduction or decorrelation [10], [11]. PCA represent data in a form that increases the mutual independence of influential components by means of Eigen-analysis.

K-Nearest Neighbor (KNN) is a simple algorithm that can store all available cases and classifies new cases based on a similarity measure. K-Nearest Neighbor (KNN) algorithm is one of the distinctive methods used in image classification [12]. Nearest Neighbor classification of objects is based on their similarity to the training sets and the statistics defined. KNN's basic idea is that if the majority of the k nearest samples of an image in the feature space belong to a certain category, the image also belongs to this category. KNN consists of two main procedures: similarity computing and k nearest samples searching. Since KNN requires no learning and training phases and avoids overfitting of parameters, it has a good accuracy in dealing with classification tasks with more samples and less classes. The k nearest neighbor (KNN) classifier is based on the Euclidean distance between a test sample and the specified training samples as described by [13].

Mathematically, if x_i is an input with some p features ($x_{i1}, x_{i2}, x_{i3}, \dots, x_{ip}$), n the total number of input samples ($i = 1, 2, \dots, n$) and p the total number of features ($j = 1, 2, \dots, p$). The Euclidean distance $d(x_i, x_l)$ between sample x_i and x_l ($l = 1, 2, \dots, n$) is defined as:

$$d(x_i, x_l) = \sqrt{(x_{i1} - x_{l1})^2 + (x_{i2} - x_{l2})^2 + \dots + (x_{ip} - x_{lp})^2} \dots (1)$$

II. REVIEW OF RELATED WORKS

In [14], the authors compared the performance of unsupervised feature learning and transfer learning against simple patch initialization and random weight initialization within the same setup. They showed that pre-training helps to train CNNs from few samples and that the correct choice of the initialization scheme can push the network's performance by up to 41% compared to random initialization. Their results show that pre-training systematically improves generalization capabilities when handling datasets with few samples. They concluded that the choice of a pre-training method depend highly on the dataset used. They used the traditional filter selection and softmax as their classifier.

In their work, [15] evaluated and compared the performance of the support vector machine (SVM) and K-nearest neighbour (K-NN) classifiers in diagnosis respiratory pathologies using respiratory sounds from R.A.L.E database. They measured the performances of the classifiers using the confusion matrix technique. They found that the K-NN classifier was better than the SVM classifier for the discrimination of pulmonary acoustic signals from pathological and normal subjects obtained from the RALE database.

The authors [16] derived a new class of fast algorithms for convolutional neural networks using Winograd's minimal filtering algorithms. The algorithms were derived for network layers with 3x3 as filter size for image recognition tasks. Their algorithms reduced arithmetic complexity up to 4 times compared with direct convolution, while using small block sizes with limited transform overhead and high computational intensity. Their work basically concentrated on the convolution layer and did not consider other layers of the ConvNet.

In [17], the authors studied the performance of different classifiers on the CIFAR-10 dataset, and build an ensemble of classifiers to reach a better performance. They were able to show that on CIFAR-10, K-Nearest Neighbors (KNN) and Convolutional Neural Network (CNN), on some classes, are mutually exclusive, and as such produced higher accuracy when combined. They applied the concept of Principal Component Analysis (PCA) to reduce overfitting in KNN, and then combined it with a CNN to increase its accuracy. Their work combined the KNN with CNN for feature extraction which is different from our proposed work.

The authors in [18] investigated a series of data augmentation techniques to progressively improve the prediction invariance of image scaling and rotation. They used SVM classifier as an alternative to softmax to enhance generalization ability. The recognition rate was up to 92.74% on the patch level with data augmentation and classifier boosting. Their results showed that the combined CNN-SVM model beats models of traditional features with SVM as well as the original CNN with softmax.

To deal with the problem associated with softmax classifier, [5] proposed a new technique of combining Biomimetic Pattern Recognition (BPR) with ConvNets for image classification. BPR performs class recognition by a union of geometrical cover sets in a high-dimensional feature space and therefore can overcome some disadvantages of traditional pattern recognition. They evaluated the method using three image datasets: MNIST, AR, and CIFAR10. We are applying their concept but using K-NN instead of the BPR.

In [6], the author presented an architecture which combines a Convolutional Neural Network (CNN) and a linear SVM for image classification. They employed the use of a simple 2-Convolutional Layer with max-pooling. They tested their architecture using Fashion-MNIST datasets and found that the CNN-softmax outperformed the CNN-SVM. They confessed that there was no enough preprocessing of the data in their study and they need to improve on their model to achieve more accurate results,

In their paper,[19] proposed a Presentation Attack Detection (PAD) method called Spoof Detection for near-infrared (NIR) camera-based finger-vein recognition system using Convolutional Neural Network (CNN) to enhance the detection ability of previous handcrafted methods. They derived a suitable feature extractor for the PAD using ConvNet. They processed the extracted image features in order to enhance the presentation attack finger-vein image detection ability of the CNN method using principal Component Analysis Method (PCA) for dimensionality reduction of feature space and Support Vector Machine (SVM) for classification. Through extensive experimental results, they endorsed that their proposed method is suitable for presentation attack finger-vein image detection and it can deliver superior detection results when compared with other ConvNets methods.

[20] presented the use of fully convolutional architectures in the notable object detection systems such as Fast/Faster RCNN to replace the fully connected layer of ConvNets. They derived a general formula specifically to accurately design the input size of the various fully convolutional networks in which the convolutional layer and pooling layer are concatenated with their strides and have proposed an efficient architecture of skip connection to accelerate the training process. They compared their model with Fast RCNN and the accuracy increased by about 2%. They tested their method using a very small data set. Again, using CNN as a classifier might not be feasible because fully connected layer is able to generalize the feature extracted into the output space and also the output need to be scalar.

In their work, [12] proposed a k-Tree method to learn different optimal k values for different test and new samples, by involving a training stage in the kNN classification. In the training stage, kTree method first learns optimal k values for all training samples by a new sparse reconstruction model, and then constructed a decision tree using training samples and the learned optimal k values. In the test stage, the kTree obtained as output the optimal k value for each tested sample, before the kNN classification was carried out using the learned optimal k value and all training samples. Their model had similar running cost but higher classification accuracy, compared with traditional kNN methods but less running cost. It also realised similar classification accuracy, compared with the newly kNN methods, which assign different k values to different test samples.

They [12] further proposed an improvement version of kTree method called k*Tree method to speed up the test stage by extra storing the information of the training.

samples in the leaf nodes of kTree, such as the training samples located in the leaf nodes, their kNNs, and the nearest neighbor of these kNNs. The model was able to conduct kNN classification using a subset of the training samples in the leaf nodes rather than all training samples used in the newly kNN methods. This actually reduced running cost of test stage.

The authors in [12] used the powerful parallel computing ability of quantum computers to optimize the efficiency of image classification. Their scheme was based on quantum K Nearest-Neighbor algorithm. The complexity of the quantum algorithm was found to be superior to the classical algorithms. Moreover, the measurement step is executed only once to ensure the validity of the scheme. Their experimental results showed that, the classification accuracy is 83.1% on Graz-01 dataset and 78% on Caltech-101 dataset, which is close to existing classical algorithms and they concluded that their quantum scheme has a good classification performance and improved the efficiency.

III.METHODOLOGY

In this work, we present an improved ConvNet model with K-NN image classification. The system framework has an automatic feature extraction using ConvNet. K-NN is adopted as the classifier exploiting the features extracted from the previous layers. To obtain the optimal coverage, PCA is used to reduce the dimension of the fully connected layer in order to overcome the time complexity of the K-NN.

A. CONVOLUTIONAL NEURAL NETWORK (ConvNet)

To extract the ConvNet based feature, network training is used first. The training of ConvNet is composed of two main procedures, namely, forward propagation and back-propagation Yang, et al., (2015).

1) FORWARD CONVOLUTIONAL LAYERS:

Suppose that we have some $M \times N$ input image, where M is the width and N the height. We train a convolutional filter ω of $m \times n$, our convolutional layer output will be of size $(M - m + 1) \times (N - n + 1)$.

To compute the pre-nonlinearity input to some unit x_{ij}^l in a layer, we need to sum up the contributions (weighted by the filter components) from the previous layer cells given by:

$$x_{ij}^l = \sum_{a=0}^{m-1} \sum_{b=0}^{n-1} \omega_{ab} y_{(i+a)(j+b)}^{l-1} \dots \dots \dots (3)$$

This is just a convolution. The convolutional layer then applies its nonlinearity:

$$y_{ij}^l = \sigma(x_{ij}^l) = \begin{cases} \alpha x & (x < 0) \\ x & (x \geq 0) \end{cases} \dots \dots \dots (4)$$

Where $\sigma(x_{ij}^l)$ is define by the Leaky ReLU activation function and α is a small positive value, usually 0.01.

Max-pooling is applied for down-sampling. The max-pooling layers are relatively simple, and do no learning themselves. They fundamentally accept a $k \times k$ region and produce a single value, which is the maximum in the region. As such, given an input $M \times N$, max-pooling will output a $\frac{M}{k} \times \frac{N}{k}$, thus a $k \times k$ region is reduced to a single value.

2) BACKWARD PROPAGATION:

We start by defining some error function, E with respect to each neuron output $\frac{\partial E}{\partial y_{ij}^l}$ applying chain rule, we can obtain the equation below:

$$\frac{\partial E}{\partial \omega_{ab}} = \sum_{i=0}^{M-m} \sum_{j=0}^{N-n} \frac{\partial E}{\partial x_{ij}^l} \frac{\partial x_{ij}^l}{\partial \omega_{ab}} \dots (5)$$

From equation (3)

$$\frac{\partial E}{\partial \omega_{ab}} = \sum_{i=0}^{M-m} \sum_{j=0}^{N-n} \frac{\partial E}{\partial x_{ij}^l} y_{(i+a)(j+b)}^{l-1} \dots (6)$$

In this case, we must sum over all x_{ij}^l expressions in which ω_{ab} occurs. In order to compute the gradient, we need to know the values of $\frac{\partial E}{\partial x_{ij}^l}$ and are computed using equation (4):

$$\frac{\partial E}{\partial x_{ij}^l} = \frac{\partial E}{\partial y_{ij}^l} \frac{\partial y_{ij}^l}{\partial x_{ij}^l} = \frac{\partial E}{\partial y_{ij}^l} \frac{\partial}{\partial x_{ij}^l} (\sigma(x_{ij}^l)) = \frac{\partial E}{\partial y_{ij}^l} (\sigma'(x_{ij}^l)) \dots (7)$$

Where $\sigma'(x_{ij}^l)$ is the partial derivative of $\sigma(x_{ij}^l)$ with respect to x_{ij}^l .

There is need to also propagate errors back to the previous layers.

$$\frac{\partial E}{\partial y_{ij}^{l-1}} = \sum_{a=0}^{m-1} \sum_{b=0}^{n-1} \frac{\partial E}{\partial x_{(i-a)(j-b)}^l} \frac{\partial x_{(i-a)(j-b)}^l}{\partial y_{ij}^{l-1}} = \sum_{a=0}^{m-1} \sum_{b=0}^{n-1} \frac{\partial E}{\partial x_{(i-a)(j-b)}^l} \omega_{ab} \dots (8)$$

where $\frac{\partial x_{(i-a)(j-b)}^l}{\partial y_{ij}^{l-1}} = \omega_{ab}$ from equation (6).

B. DIMENSION REDUCTION BASED ON PRINCIPAL COMPONENT ANALYSIS (PCA)

The classic approach to PCA is to perform the eigen-decomposition on the covariance matrix C , which is a $m \times m$ matrix where each element represents the covariance between two features. Reducing data from m -dimensional to k -dimensional, we first compute the covariance matrix

$$C = \frac{1}{m} \sum_{i=1}^n (x^{(i)})(x^{(i)})^T \dots (9)$$

$x^{(i)}$ is an $m \times 1$ matrix. We then find eigenvectors of matrix C by using the Singular Value Decomposition (SVD)

$$SVD(C) = [U, S, V] \dots (10)$$

U is an $m \times m$ real or complex unitary matrix, also called the left-singular vectors of C are a set of orthonormal eigenvectors of CC^* , S is an $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal, and V is an $n \times n$ real or complex unitary matrix also known as the right-singular vectors of C are a set of orthonormal eigenvectors of C^*C . U matrix is also a matrix of size $m \times m$ and it corresponds to the eigenvectors needed for the computation. To reduce a system from m -dimensions to k -dimensions, we will take the first k -vectors from U which corresponds to the first k columns of matrix U :

$$U = \begin{bmatrix} | & | & & | \\ u^{(1)} & u^{(2)} & \dots & u^{(m)} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{m \times m}$$

We then take first k columns of the u matrix and stack in columns $m \times k$. k is chosen in terms of the ratio between averaged squared projection error with total variation in data:

$$\frac{\frac{1}{m} \sum_{i=1}^m ||x^{(i)} - x_{approx}^{(i)}||^2}{\frac{1}{m} \sum_{i=1}^m ||x^{(i)}||^2} \leq 0.01 \dots (11)$$

This means we will have to retain 99% of the variance because we will have to lose a very little information in the dimensionality reduction, so that when we decompress we can regenerate the same data. The reduced matrix is called U_{reduce} which is a $k \times n$ in order to change x , which is m -dimensional to z which is k -dimensional.

$$z = (U_{reduce})^T * x \dots (12)$$

This means $[k \times n] * [n \times 1] = k \times 1$ matrix.

C. CLASSIFICATION USING THE K-NEAREST NEIGHBOR (K-NN)

The classification rules of KNN are generated by the training samples themselves without any additional data. KNN classification algorithm predicts the test sample's category according to the K training samples which are the nearest neighbors to the test sample, and judge it to that category which has the largest category probability. The process of KNN algorithm to classify an image X according to [22] is:

Suppose that there are j training categories as C_1, C_2, \dots, C_j , and the sum of the training samples is N . After applying PCA, the image becomes k -dimension feature vector, we make image X to be the same vector form (X_1, X_2, \dots, X_m) as the training samples. We then calculate the similarities between all training samples and image X , taking the i^{th} image $d_i = (di_1, di_2, \dots, dim)$, the similarity $SIM(X, d_i)$ as

$$SIM(X, d_i) = \frac{\sum_{j=1}^m X_j \cdot d_{ij}}{\sqrt{\sum_{j=1}^m X_j} \cdot \sqrt{\sum_{j=1}^m d_{ij}}} \dots \dots \dots (13)$$

We choose k samples which are larger from N similarities of $SIM(X, d_i), (i 1, 2, \dots, N)$ and treat them as a KNN collection of X . Then, calculate the probability of X belong to each category respectively with the following formula.

$$P(X, C_j) = \sum_{d_j \in KNN} SIM(X, d_i) \cdot y(d_i, C_j) \dots \dots \dots (14)$$

Where, $y(d_i, C_j)$ is a category attribute function, which satisfies:

$$y(d_i, C_j) = \begin{cases} 1, & d_i \in C_j \\ 0, & d_i \notin C_j \end{cases} \dots \dots (15)$$

Image X is classified based on the category which has the largest $P(X, C_j)$.

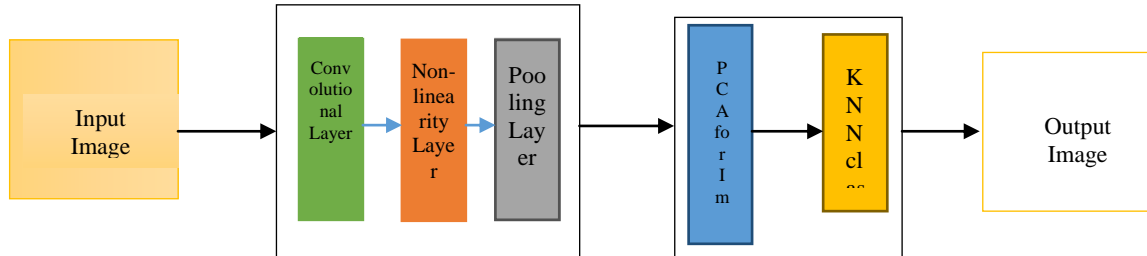


Fig 1: The Proposed Model Architecture

IV. THE PROPOSED SYSTEM

In this work, we are adopting the concept of replacing the softmax classifier as in the work of [5] with a simpler classifier K-Nearest Neighbor (K-NN).

We first use the ConvNet for feature extraction so that we can produce filtered feature maps. Our work will also adopt the use of Principal Component Analysis (PCA) to reduce the dimension of the fully connected layer of the ConvNet before input into the K-NN for classification. We adopt the use of PCA so that we reduce the computational time of the K-NN and to avoid overfitting.

A. ARCHITECTURE OF THE PROPOSED SYSTEM

A ConvNet architecture with alternating convolutional and max-pooling layers is used here. The nodes in the output layer matches one character class. After ConvNet training, PCA is applied to the feature vector for dimension reduction before being fed into K-NN for classification.

B. PERFORMANCE METRICS

The performance of the proposed algorithm is to be evaluated using the confusion matrix by calculating:

1) THE PRECISION-RECALL CURVE.

Precision is defined as the number of true positive detections retrieved, divided by the total number of true and false positives retrieved. Recall is defined as the number of true positive detections retrieved, divided by the total number of positive samples known from the original data.

Mathematically:

$$Precision = \frac{(TP)}{(TP) + (FP)} \dots (16)$$

$$Recall = \frac{(TP)}{(TP) + (FN)} \dots (17)$$

2) INTERPOLATED AVERAGE PRECISION

Interpolated average precision of the detections for each class, similar to the pascal VOC challenge. An interpolated (monotonically decreasing) curve is computed for visualization purposes and for calculating the interpolated average precision. The interpolated curve is created by replacing the actual precision values with maximum precision values from the remaining part of the curve. Regularly, average precision is calculated by computing a discrete approximation of the area under the precision-recall curve (by calculating a cumulative sum of all precisions multiplied by recall delta-values). In the PASCAL Visual Object Class (VOC) challenge, interpolated average precision is used instead, which reduces the impact of perturbations caused by slight ordering differences [23].

The interpolated average precision P_{interp} at a certain recall level r is defined as the highest precision found for any recall level $r' \geq r$:

$$P_{interp}(r) = \max_{r' \geq r} p(r') \dots (18)$$

The confusion matrix is defined as:

Table I: Confusion Matrix [15]

		Predicted Class	
		Positive(P)	Negative(N)
Actual Class	P	True Positive (TP)	False Negative (FN)
	N	False Positive (FP)	True Negative (TN)

Positive: Observation is positive

Negative: Observation not positive

True Positive: Observation is positive and is predicted positive

True Negative: Observation is negative and is predicted negative

False Positive: observation is negative but predicted positive

False Negative: Observation is positive but predicted negative

V. CONCLUSION

This paper tends to design an improved Convolutional Neural Networks (ConvNets) with improved precision and accuracy. We employ the use of Principal Component Analysis (PCA) to reduce the dimension of the image and K-Nearest Neighbor (KNN) for classification. When successfully implemented, the proposed system should be able to accurately classify images based on a given training set and test set. We will evaluate our work with the work of [5] and [6] in terms of accuracy.

REFERENCES

- [1] A. Karpathy, "Convolutional Neural Networks for Visual Recognition," *Stanford Univ.*, pp. 1–21, 2016.
- [2] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [3] W. Li, S. Manivannan, S. Akbar, J. Zhang, E. Trucco, and S. J. McKenna, "Gland segmentation in colon histology images using hand-crafted features and convolutional neural networks," *softheIEEE13thInternationalSymposium onBiomedicalImaging:FromNanotoMacro(ISBI'16)*, pp. 1405–1408.
- [4] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, vol. 07–12–June, pp. 1–9.
- [5] L. Zhou, Q. Li, G. Huo, and Y. Zhou, "Image classification using biomimetic pattern recognition with convolutional neural networks features," *Comput. Intell. Neurosci.*, vol. 2017, 2017.
- [6] A. F. Agarap, "A Neural Network Architecture Combining Gated Recurrent Unit (GRU) and Support Vector Machine (SVM) for Intrusion Detection in Network Traffic Data," 2017.
- [7] Y. Tang, "Deep Learning using Linear Support Vector Machines," *Deeplearning.Net*, 2013.
- [8] R. A. Nugraheeni and K. Mutijarsa, "Comparative Analysis of Machine learning KNN, SVM and Random Forest Algorithm for Facial Expression Classification," in *International Seminar on Application for Technology of Information and Communication*, 2016.
- [9] G. Gan, Yanhai; Jun, Liu; Dong, Junyi; Zhong, "A PCA-Based Convolutional Network," no. May, 2015.
- [10] R. C. Gonzales and R. E. Woods, *Digital Image Processing*. 2002.
- [11] H. H. Barret, *Foundations of Image Science*. 2004.
- [12] Y. Dang, N. Jiang, H. Hu, Z. Ji, and W. Zhang, "Image Classification Based on Quantum KNN," 2018.
- [13] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Adv. neural Inf. Process. Syst.*, pp. 1473–1480, 2006.
- [14] R. Wagner, M. Thom, R. Schweiger, and A. Rothermel, "Learning Convolutional Neural Networks From Few Samples," pp. 1884–1890, 2013.
- [15] R. Palaniappan, K. Sundaraj, and S. Sundaraj, "A Comparative study of SVM and KNN Machine Learning Algorithms for the Respiratory Diagnosis of Pathologies using Pulmonary Acoustic Signals," *BMC Bioinformatics*, 2014.
- [16] A. Lavin and S. Gray, "Fast Algorithms for Convolutional Neural Networks," in *Proc. IEEE CVPR*, 2016, pp. 4013–4021.
- [17] Y. Abouelnaga, O. S. Ali, H. Rady, and M. Moustafa, "CIFAR-10 : KNN-based Ensemble of Classifiers," 2016.
- [18] D. Xue, R. Zhang, and H. F. Y. Wang, "CNN-SVM for Microvascular Morphological Type Recognition with Data Augmentation," *J. Med. Biol. Eng.*, vol. 36, no. 6, pp. 755–764, 2016.
- [19] D. T. Nguyen, H. S. Yoon, T. D. Pham, and K. R. Park, "Spoof detection for finger-vein recognition system using NIR camera," *Sensors (Switzerland)*, vol. 17, no. 10, 2017.
- [20] Y. Ren, "Object Detection Based on Fast / Faster RCNN Employing Fully Convolutional Architectures," vol. 2018, 2018.
- [21] J. Yang, J. Gao, G. Wang, and S. Zhang, "Natural Scene Recognition Based on Superpixels and Deep Boltzmann Machines," in *arXiv:1506.07271 [cs]*, 2015, no. 100, pp. 2369–2374.
- [22] Y. Lihua, D. Qi, and G. Yanjun, "Study on KNN Text Categorization Algorithm," *Micro Comput. Inf.*, no. 21, pp. 269–271, 2006.
- [23] M. Everingham, L. Van-Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes Challenge," *Ijcv*, vol. 88, no. 2, pp. 303–338, 2010.