A DEEP LEARNING IMAGE CLASSIFICATION USING TENSORFLOW FOR OPTICAL AVIATION SYSTEMS

A. Acet and A. E. Akkaya

***Abstract*—** Deep learning has become very popular in recent years. Great progress has been made in the task of classifying images with the development of deep learning. This research utilized the deep learning methods in TensorFlow to classify the bird and airplane images. In the first step, a general framework for the classification of deep learning images, an image classification network namely airplane images and bird images are built. Then, the images were randomly chosen from the Caltech-UCSD Birds-200-2011 and Caltech 101 datasets. To correctly classify airplane and bird images, a total of 1600 images used. The 1072 images used to train the network and the 528 images used to test built deep learning network. The training phase lasts only 20 epochs to achieve 100% accuracy on the train set. The test data were classified as 99.05%. Overall accuracy is 99.69%. This research has a certain importance to explore the use of cognitive systems approach in aviation safety.

***Keywords*—** *Deep learning, TensorFlow, CUDA, Image classification.*

1. INTRODUCTION

IN RECENT years deep learning has become a hot topic of research. In the area of artificial intelligence, image recognition, pattern recognition and autonomous driving deep learning have made significant progress. Deep learning has some benefits. These networks are independent of artificial features. Also, deep learning systems can learn adaptively. Thus the algorithm's reliability and versatility are greater than the conventional approach of image processing. Convolutional Neural Networks (CNNs) are a special type of deep learning models widely used in areas such as image classification [1] and natural language processing [2]. Before deep learning architectures, conventional artificial neural networks were used to determine what features an image contains. To achieve this, the image is transformed into a column or row vector and given to the system input. In this type of structure, many features are composed of side-by-side pixel values. However, the human perception system looks at corners, lines and rounded shapes to perceive an image. When an image is flattened and brought into a row or column vector, all the fine details disappear. It is almost impossible to try to solve this structure, which is difficult to perceive even by perfectly functioning human intelligence, using machine learning techniques. CNNs has been developed as a solution to this problem. CNNs use feedforward structure. Unlike conventional artificial neural networks, it has convolution and pooling layers for feature extraction and reducing the size of the input image, respectively. Using both layers, important features in the image can be extracted.

Ayça ACET. Inonu University, Malatya, Turkey, (a.aycaacet@gmail.com)

Abdullah Erhan AKKAYA. Inonu University, Malatya, Turkey, (abdullahakkaya@gmail.com)

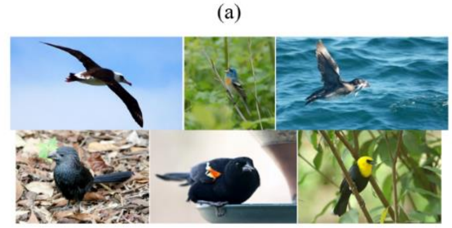
Manuscript received May 12, 2020; accepted May 29, 2020.
Digital Object Identifier:

This research utilized the deep learning methods in TensorFlow [3] to classify the bird and airplane images. Firstly, the residual network, a general framework for the classification of deep learning images, an image classification network namely airplane images and bird images are built. Then, the Caltech-UCSD Birds-200-2011 [4] and Caltech 101 [5] datasets are used to train and validate the neural network. To correctly classify airplane and bird images, total of 1600 images consists of birds and airplanes used. The split rate of the network is 0.33. The 67% of images (1072 images) used to train the network. The %33 of the images (528 images) used to validate the network. The training phase lasts only 20 epochs to achieve 100% correctness and the test data were classified as 99.4%.. Although they have similar structures, aircraft and bird images have been successfully distinguished from each other. This research has a certain importance to explore the use of cognitive systems approach in aviation safety.

2. DATASET

Caltech-UCSD Birds-200-2011 [4] and Caltech 101 [5] datasets used for train and test steps. Total of 1600 images is chosen from the dataset that has various sizes. These images are labelled as airplane image, and the bird image (Figure 1). All dataset images are divided into two sets. The training set consists of 1072 images and the test set consists of 528 images. Firstly, to train the weights of the built CNN, a training set is used. The validation set is randomly chosen from the set to confirm the model's generalization ability. Therefore, the model weights are selected to be saved on the validation data according to the value of the loss function. Finally, at the test step, the saved weights of the trained model are used to give decisions about the class on the test set.

(a)



(b)

Fig.1. Sample airplane (a) and bird (b) images used for a classification task.

 ISSN: 2548-0650

3. TOOLS AND NETWORK MODEL

3.1. Development Environment

In this study, CNN using TensorFlow application was performed on GPU using Python language in Anaconda environment. The calculations on the GPU have been performed via the Nvidia CUDA library. CUDA acceleration package and *cudacnn* are installed, and then some related OpenCV-Python, TensorFlow, Scikit-learn and Keras packages are installed. Proposed network has been trained on the Nvidia GeForce GTX 1070 8GB 256-bit graphic card that has 1920 CUDA cores.

Python programming language provides an easy way to solve the problems with coding flexibility [6]. Python is a platform-independent, less time-consuming scripting language [6]. It is a high level interpreted. Many developers use Python for developing easy to code programs. ABC language was popular but shortly after arriving on the market Python took its crown. Python language is a dynamic language and it has garbage collection mechanism.

The Scikit-learn library is widely using for modelling with data mining and data analysis. The Scikit-learn, NumPy, matplotlib, and SciPy libraries contain simple tools for machine learning classification, regression and clustering tasks [7]. Supervised learning algorithms, non-supervised learning algorithms, feature extraction and cross-validation are some features of this library.

Sample Python code for proposed CNN is given below.

```
model.add(Convolution2D(kernel_size=3, strides=1,
filters=32, padding='same', activation='relu', name='conv1',
input_shape=input_shape))

model.add(MaxPooling2D(pool_size=2, strides=2))
…
model.add(Dense(num_classes, activation='softmax'))

model.add(Dropout(0.8))

model.add(Dense(num_classes, activation='softmax'))

optimizer = adam(lr=0.001, beta_1=0.9, beta_2=0.999,
epsilon=1e-08, decay=0.0)

model.compile(optimizer=optimizer,
loss='categorical_crossentropy', metrics=['accuracy'])

hist = model.fit(X_train, y_train, batch_size=64,
epochs=num_epoch, verbose=1, validation_data=(X_test,
y_test), shuffle=True)
```

3.2. Convolutional Neural Networks (CNNs)

CNNs are feedforward neural networks. In feedforward neural networks, the signal flows over a network without loops [8]. A typical CNN model consists of convolutional, activation, pooling, fully connected, and output layers. The convolutional layer has a function composed of multiple convolutional kernels. Each kernel symbolizes a linear function in matching kernel [8]. In the pooling layer, a layer by layer down-sampling non-linear function used for aiming at reducing progressively the size of the feature representation. A fully connected layer can be considered a type of convolutional layer. The kernel size of fully connected layers is 1×1. To compute the probabilities of input image belonging to which classes, the output or prediction layer is often used at the last fully connected layer.

The proposed CNN consists of eight layers. The first, third and fifth layers are the convolution layers where the basic features of the image detected. The second, fourth and sixth layers are the pooling layers, which reduces the image size by half. In convolution layers, more detailed information about the image was tried to be obtained by using 32,64 and 128 filters of size 3x3 in first, third and fifth layers respectively. The seventh layer is a fully connected (dense) layer in which all neurons are connected together. Immediately after this layer, unnecessary nerve cells were deleted using a 0.5 dropout value in order to prevent overfitting problem. The eight and last layer determines which results will be included in the class of airplanes or birds from the previous layers. One of the activation functions the Rectified Linear Unit (ReLU) [9] is used in the first, third, and fifth layers of the convolution layers. The ReLU function used in conjunction with other layers. ReLU activation function has become the most commonly used in deep learning networks and more popular than logistic sigmoid and hyperbolic tangent functions [10]. In this study, ReLU function used as an activation function. The ReLU is, where x is the input value to the ReLU function, formulated as follows,

$$f(x) = max(0, x) \tag{1}$$

It truncates all negative values input to zero. Only half of the ReLUs is activated when used in combination with a batch normalization layer. Therefore at a given time, it results in sparse activations. The Softmax activation function used in the eight-layer.

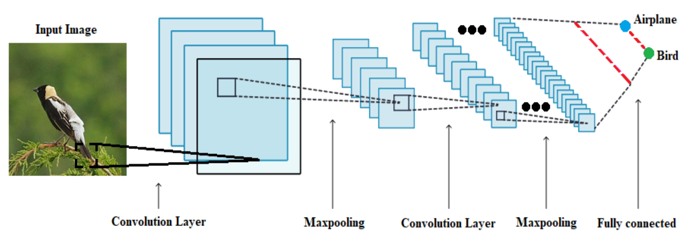


Fig.2. Framework of proposed CNN.

TABLE I
PARAMETERS OF PROPOSED CNN

Layer name	Layer Parameters
Input Image	128x128 pixels
Conv1	Kernel size=3, strides=1, filters=32x32, ReLU
Maxpooling	Pool size=2, strides=2
Conv2	Kernel size=3, strides=1, filters=64x64, ReLU
Maxpooling	Pool size=2, strides=2
Conv3	Kernel size=3, strides=1, filters=128x128, ReLU
Maxpooling	Pool size=2, strides=2
Dense layer	Neuron size=128, ReLU, Dropout=0.8
Output	Average pooling, fully connected 2 class, Softmax

3.3. Categorical Cross-Entropy Loss Function

Categorical Cross-Entropy Loss Function, also called Softmax Loss function, consists of Softmax activation and Cross-Entropy loss functions. When this function used, CNN has trained to output a probability over the classes for each image. Equation (2) and Equation (3) shows Softmax output and Categorical Cross-Entropy loss functions.

$$f(S)_i = \frac{e^{S_i}}{\sum_j^C e^{S_j}}. \tag{2}$$

$$CE = -\sum_i^C t_i log(f(s)_i). \tag{3}$$

4. Train and Test CNN

4.1. Train the CNN Architecture

Adam [11] is selected as an optimization algorithm. The initial learning rate of Adam algorithm is set to 0.001. The batch size is chosen 64 for the Nvidia GTX1070 graphic card. The network model has trained 20 epochs. At the training step, data augmentation is used by the function generator in Keras. The model checkpoint class is used to select optimal model weights with respect to the validation loss value. Figure 3 shows train loss and validation loss values for 20 epochs. Figure 4 show train accuracy and test accuracy curves.

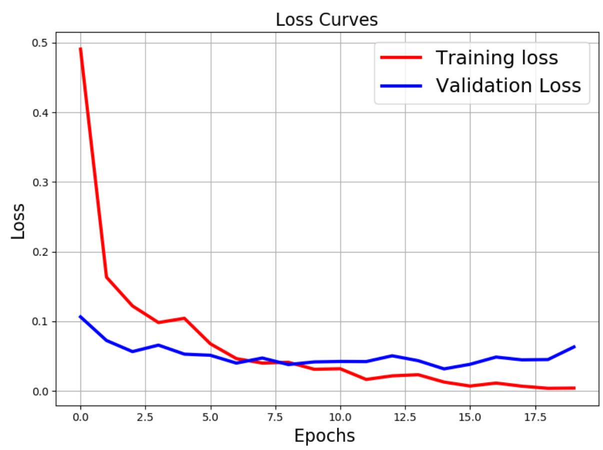


Fig.3. The train and validation loss curves of proposed CNN.

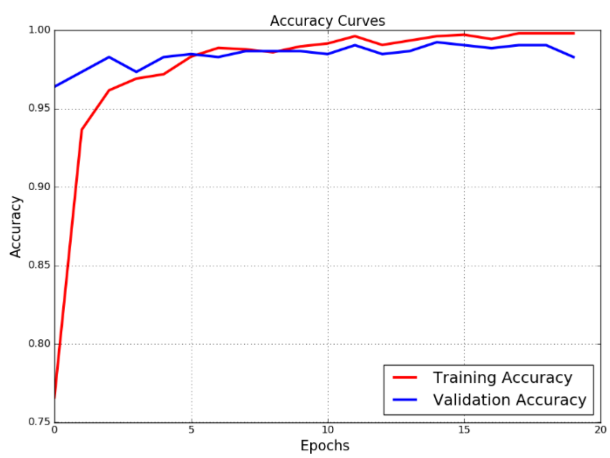


Fig.4. The train and validation accuracy curves of proposed CNN.

4.2. Test Result

After the CNN trained, the generalization ability of the trained model should be evaluated. To achieve this the test set which consists of 240 airplane images and 288 bird images is used. For the evaluation of the proposed CNN, we assumed the class containing airplane images as the positive class, and therefore the accuracy, sensitivity, and specificity are computed with reference to this. We report the definitions for these parameters, just for clarity.

- True Positive (TP) is the number of airplane images correctly classified;
- True Negative (TN) is the number of bird images correctly classified;
- False Positive (FP) is the number of bird images incorrectly classified as airplane image;
- False Negative (FN) is the number of airplane images incorrectly classified as bird image.

$$Accuracy = \frac{TP + TN}{(TP + TN) + (FP + FN)} = 1 - error$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP}$$

$$False\ positive\ rate = \frac{FP}{TN + FP} = 1 - specificity$$

Table II shows true and predicted labels after the prediction step on the test set. The sensitivity and specificity are two indexes generally used to evaluate the performance of the classifier. Table II is also called the confusion matrix.

TABLE II
TEST RESULTS

		Predicted label	
		Airplane (Positive)	Bird (Negative)
True label	Airplane (True)	237 (True Positive)	3 (False Negative)
	Bird (False)	2 (False Positive)	286 (True Negative)

Error rate and accuracy are can be obtained from the confusion matrix. The error rate is calculated by dividing the number of all false estimates by the total number of the data set. The best worst error rate is 1, the best is 0. Accuracy is calculated by dividing the number of all correct estimates by the total number of the data set. The worst accuracy is 0, the best is 1. It can be calculated with (1–error rate). Sensitivity is calculated by dividing the number of true positive predictions by the total positive number. This is called a recall or true positive rate (TPR). The worst sensitivity is 0, the best is 1. Specificity is calculated by dividing the number of correct estimates by the total number of negatives. This is also called true negative rate (TNR). The worst specificity is 0, the best is 1. Sensitivity is calculated by dividing the number of true positive predictions by the total positive predictions. This is called a positive predictive value (PPV). The worst precision is 0, the best is 1. The false-positive ratio is calculated by dividing the number of false-positive predictions by the total number of negatives. The worst false positive rate is 1, the best is 0. It can also be calculated as (1–specificity).

Table III shows the discrimination ability of the proposed convolutional neural network overtraining and testing steps. Also, whole datasets are shown in the table in terms of accuracy, sensitivity and specificity. The whole dataset split by 0.33 split ratio.

TABLE III
DISCRIMINATING ABILITY OF THE PROPOSED CNN CLASSIFIER

	Training Set	Testing Set	Whole Data
Accuracy	100%	99.05%	99.69%
Sensitivity	100%	98.75%	99.59%
Specificity	100%	99.30%	99.77%
Precision	100%	99.16%	99.72%

Figure 5 is an example for wrongly classified and labelled airplane image with 97.42% probability. This image is a bird image and for the human optical system, it has different properties. CNN decreases this image size at every max-pooling step. If we zoom out enough we see it looks like a plane because of the wingspan.

Fig.5. Misclassification of bird image at proposed CNN.

5. CONCLUSIONS

In this paper, based on binary image classification, airplane and bird classification model is built. Then, equally chosen 1600 airplane and bird images are divided into two sets, train and test. The optimal model weight selected is done by the loss value of the validation set. Finally, the accuracy of the trained model has 99.16% precision and 98.75% sensitivity ratios. This study valuable for exploring the application of deep learning method in optical early bird detection systems for aviation.

ACKNOWLEDGMENT

The study is selected from Congress Proceedings Abstract Book, International Conference on STEM And Educational Sciences, Muş Alparslan University, May, 2018.

REFERENCES

[1] D. Ciresan, U. Meier, and J. Schmidhuber, “Multi-Column Deep Neural Networks for Image Classification,” Technical Report, arXiv:1202.2745, 2012.

[2] R. Collobert and J. Weston, “A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning,” Proc. Int’l Conf. Machine Learning, 2008.

[3] Martín Abadi et. al., “TensorFlow: Large-scale machine learning on heterogeneous systems”, 2015. Software available from tensorflow.org.

[4] Wah C., Branson S., Welinder P., Perona P., Belongie S. “The Caltech-UCSD Birds-200-2011 Dataset.” Computation & Neural Systems Technical Report, CNS-TR-2011-001.

[5] L. Fei-Fei, R. Fergus and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. IEEE. CVPR 2004, Workshop on Generative-Model Based Vision. 2004.

[6] AlizaSarlan, ChayanitNadam, ShuibBasri ,”Twitter Sentiment Analysis”.

[7] KUNAL JAIN,2015,”Scikit-learn in Python – the most important Machine Learning tool I learnt last year!”. Online https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/ , Accessed:2020-04-29.

[8] B. Xu, N. Wang, T. Chen, M. Li, Empirical evaluation of rectified activations in convolutional network, arXiv preprint arXiv:1505.00853 , (2015).

[9] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in Proceedings of the International Conference on Artificial Intelligence and Statistics, pp. 315–323, 2011.

[10] A. Karpathy, “Commonly used activation functions.” http://cs231n.github.io/linear-classify/#loss. Accessed:2020-04-26.

[11] D. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980 , (2014).

BIOGRAPHIES

Ayca Acet received the BSc degree in Computer Engineering from TOBB University of Economics and Technology, Ankara, Turkey, in 2017. She currently continues M.S. degree in Computer Engineering, at Inonu University, Malatya, Turkey. Her general research interests include machine learning methods, deep learning and data mining.

Abdullah Erhan Akkaya received the BSc, M.S. and Ph.D. degrees, in Computer Engineering from Firat University, Elazig, Turkey, in 2007, Erciyes University, Kayseri, Turkey, in 2012 and Inonu University, Malatya, Turkey in 2017 respectively. He is currently an Assistant Professor of Computer Engineering, at Inonu University, Malatya, Turkey. His general research interests include machine learning methods, image processing, sensor fusion and cognitive systems.