ISCTJournal ISCEA
International Supply Chain Technology Journal
ISSN 2380-1204

https://doi.org/10.20545/isctj.v07.i07.02

Image Classification Using Deep Learning Architecture – EfficientNET for COVID-19 Medicine Consumption

Shaik, M.S[1], Azeem, G[2] Jones, E.C[3]
[1] *University of Texas at Arlington, USA.* [2,3] *University of Texas at Arlington, USA.*
[1]mohammedsuraj.shaik@uta.edu, [2]gohar.azeem@uta.edu, [3]ecjones@uta.edu

Abstract - The Coronavirus disease 2019 (COVID-19) pandemic has impacted the world like no other pandemic in history. COVID-19 has progressed rapidly affecting health care in the global community. In this paper, we study the Image classification approach for confirming the consumption of medicine by people. The motivation for this research is to reduce contact with the people, without having them in physical observation and help minimize the spread of COVID-19. We used EfficientNet, a Deep Learning Convolution Neural Network (CNN) [7] architecture for systematical use of model scaling and balancing network depth, width, and resolution for better performance of our model. In this paper, we used the data augmentation technique to avoid overfitting. We used image hashing to compare and remove relatively similar images as data cleaning step and demonstrate the effectiveness of this approach by observing the classification accuracy. We used cloud computation to train the model. This trained model will be helpful to achieve the goal. In this study, we developed a model that could successfully detect the image and confirm the pill consumption by patient with high accuracy rate. This study will help medical professionals, especially during COVID-19 pandemic, in remotely monitoring the patient's dosage and determine appropriate action in case of non-compliance.

***Keywords* – COVID-19, Deep Learning, CNN, EfficientNet, Image Classification, Image Hash, Data Cleaning.**

INTRODUCTION

Image classification is a supervised learning problem where we define a set of target classes and train a model to recognize the classes using labeled training image data. EfficientNet is a newly designed Convolution neural network (CNN) architecture by Google [3] that has set new records for both accuracy and computational efficiency. CNN is a class of deep neural networks and most commonly used to analyze images. Deep learning [8] is part of a broader family of machine learning methods based on artificial neural networks with representation learning and it is a large neural network. We use EfficientNet which is a Deep learning architecture for our Image classification problem. The process of examining the pixel values and generating a hash value that will uniquely identify an input image is Image hashing [10]. Creating modified versions of images to artificially expand the size of a given training dataset is an Image data augmentation technique [11]. Image data augmentation [11] can be achieved by ImageDataGenerator class from Keras [12] deep learning library which provides the capability to fit the given model. Keras is a neural-network library written in Python which can run on top of TensorFlow and it is opensource. Google Colab [6] is used to write and execute Python in the browser with zero configuration, free access to GPU, and easy sharing.

BACKGROUND

In previous research work, EfficientNet which is a Deep Convolutional Neural Network architecture has been used for multiclassification image problem to classify skin lesions [4]. In our paper, we used EfficientNet to classify human action recognition for the multi-classification problem.
The researchers discussed that Deep Learning models are very helpful to gain insights about the key decision-making activities for COVID-19 related work [5]. In our paper, we used EfficientNet a Deep Learning architecture to provide a contactless solution to minimize the spread of COVID-19 [9].
In this paper [3] they have compared different Convolutional Neural Networks with EfficientNet and propose that EfficientNet is a good scaling method that uniformly scales all dimensions of depth/width/resolution using a simple yet highly effective compound coefficient which achieve much better accuracy and efficiency than previous ConvNets.

METHODOLOGY

3. Data Processing

In this section, we will discuss the data collection and data cleaning steps performed.

3.1 Data Collection

We have collected data from each person for 4 major different positions Reading (fig 1a), Taking Medicine (fig 1b), Sitting (fig 1c), and Standing (fig 1d).

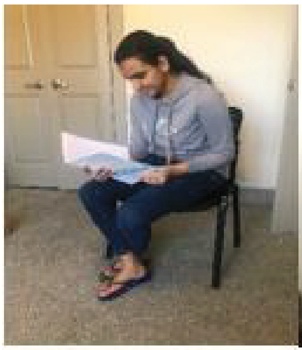


Figure 1a

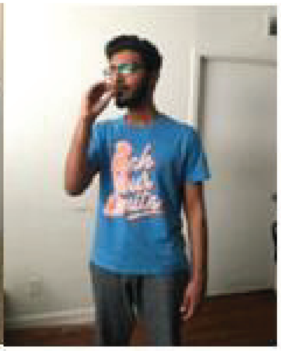


Figure 1b

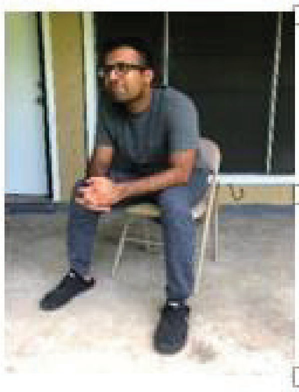


Figure 1c

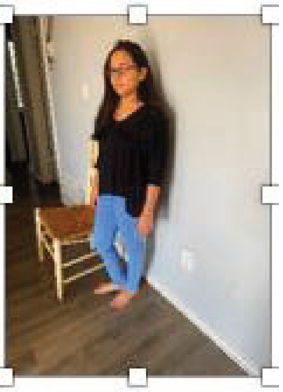


Figure 1d

We have collected some images with much relative similarity and added it to our dataset because in the future we might get almost similar data from users and our algorithm should not be biased by giving deep neural network additional opportunities to learn patterns specific to the duplicates while it is predicting. As it negatively affects the ability of our model to generalize to new images outside of training data.

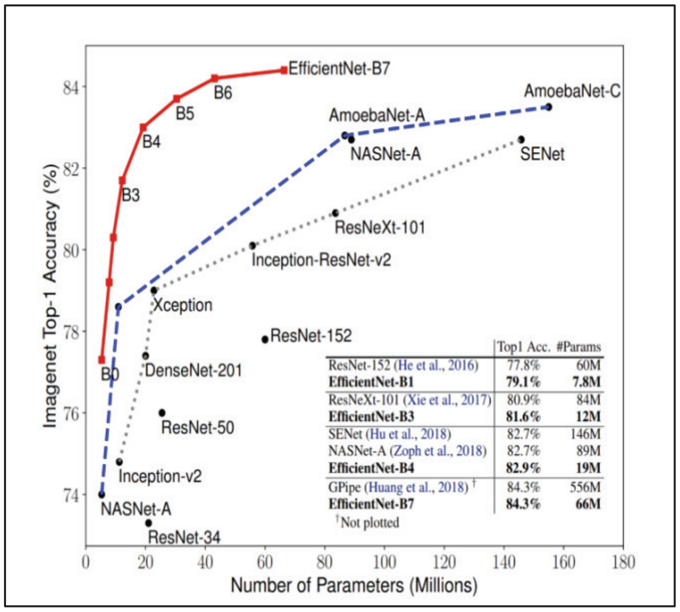
3.2 Data Cleaning (Similar Image Removal)

The dataset collected needs to be cleaned to remove the relatively similar Images. We have written a python script in Google Colab to remove such images. Used imutils [1] to grab the file paths to all images in our dataset, NumPy for image stacking, and OpenCV for image Input/Output manipulation and display [1]. Apply hashing function to every image in the dataset. Hashing function handles the calculation to create a numerical representation of the image. Images with the same hash are considered duplicates. We pass the image and hash size as parameters to the function which calculates the hash [2]. The Hash function first converts the given image to a single-channel grayscale image, then resize the image according to the hash size provided and compute the relative horizontal gradient between adjacent column pixel which is known as "difference image" [1]. Once the dataset is cleaned, we combined all the images of every person for each position. Now there are 4 different folders each representing one of four different positions that have all the images collected except the similar ones. We used this cleaned dataset to train our deep learning model using EfficientNet Architecture.

4. Model Training and Testing.

EfficientNet a deep learning architecture is used to train the model as it outperforms other Convolution Neural Networks. Convolutional Neural Networks are commonly developed at a fixed resource budget and then scaled up for better accuracy if more resources are available [3]. In this paper [3], they systematically study model scaling and identify that carefully balancing network depth, width, and resolution can lead to better performance. Based on this observation, they propose a new scaling method that uniformly scales all dimensions of depth/width/resolution using a simple yet highly effective compound coefficient.

In the Figure 2 Model size vs ImageNet Accuracy is plotted. All numbers are for single-crop, single-model. EfficientNets significantly out-perform other Convolutional Neural Networks. [3] Model is trained on NVIDIA Tesla K80 GPU Using Google Colab.



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.1%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.6%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.9%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.3%**	**66M**

†Not plotted

Dataset is divided into 4 different folders each consisting of an action or position like

Standing, Sitting, Reading, Taking Medicine. This dataset is uploaded to google drive for cloud computation. Data augmentation [11] is used to increase the size of the dataset. The total Dataset size is 2340. The test data size is 33% of the augmented dataset size.

Hyperparameters are the variables that determine the network structure and how the network is trained. Hyperparameters of the EfficientNet are tunable to get the desired result and the used parameters for our model are as follows Drop out = 0.55. Drop out is a regularization technique to reduce overfitting [13], [14] in neural networks and prevent complex co-adaptations on train data, Optimizer used is Adam. Optimizers are algorithms or methods used to change the attributes of a neural network such as weights and learning rate to reduce the losses, Learning Rate = 0.01. Learning rate [15] is a parameter used to tune an optimization algorithm which determines the step size at each iteration while moving towards a minimum of a loss function, Loss = Categorical Cross Entropy [16].

Model learns through a loss function. It's a method of evaluating how well a specific algorithm is used to model the given data, Batch size is 32 (divided the augmented train set into 32 parts). Batch size is the number of training examples used in one iteration. Architecture Details are as follows Total parameters = 4,394,280, Trainable parameters = 4,352, 257, Non-Trainable parameters = 42,023 Parameters are weights that are learnt during training and they contribute to model's predictive power.

Epoch is a number and it tells the number of times all of the training vectors used once to update the weights. For batch training, all of the training samples pass through the learning algorithm simultaneously in one epoch before weights are updated. Our model is trained for 10 epochs.

Validation Accuracy achieved for the 10th epoch is 99.87% with a loss of 0.0263%.

Code for Defined Model Architecture:

```
model = Sequential()
model.add(base(include_top=False, weights
='imagenet'))

for layer in model.layers:
   layer.trainable = True

model.add(GlobalAvgPool2D(name='global
_avg_pool_end'))
model.add(Dense(256, activation='relu', na
me='dense1_1024'))
model.add(Dropout(0.55))
model.add(Dense(64, activation='relu', name
='dense2_512'))
model.add(Dropout(0.55))
model.add(Dense(5, activation='softmax', na
me='final_prediction'))

optimizer = Adam(lr=0.01)
loss = "categorical_crossentropy"
model.compile(optimizer = 'Adam',
          loss=loss,
          metrics=['accuracy']
          )
model.summary()
```

Code for Data Augmentation and Training:

```
X_train, X_test, y_train, y_test = train_test_
split(data, label, test_size=0.33, random_stat
e=42)



datagen = ImageDataGenerator(
   featurewise_center=True,
   featurewise_std_normalization=True,
   rotation_range=20,
   width_shift_range=0.2,
   height_shift_range=0.2,
    horizontal_flip=True)

datagen_flow_train = datagen.flow(X_train,
y_train, batch_size= 32)

datagen_flow_val = datagen.flow(X_test, y_
test, batch_size= 32)

model.fit_generator(datagen_flow_train,
                epochs=10,
                validation_data=datagen_flow_
val
                )
```

5. Results and Discussion

The main objective of our model is to classify the images and identify if the person is taking medicine or not. This will help medical professionals in remotely monitoring the patient's dosage and determine appropriate action in case of non-compliance. Out of the 10 Epochs the results were consistent right from the second Epoch which tells the model trained is very successful in classifying. We achieved 99.87% of validation accuracy with the loss of 0.0263% which is a huge success. We can use this in the real world to observe the COVID-19 effected patient's recovery status and the effect of medicines on them without having them in physical observation. To achieve it we need to have the patients to be in a closed environment with cameras surrounding them which can be very helpful not only for COVID-19 but also for future pandemics and other dangerous contagious diseases.

6. Conclusion

In this paper we proposed a contactless solution of monitoring patient's pill consumption process. This study is highly significant in current scenario of COVID-19 Pandemic where there is a need of social distancing [9]. In order to achieve our objective, we created a dataset with 4 major actions related to COVID-19. An EfficientNet based deep learning model is trained on this dataset to classify human action (medicine consumption) and solve multi-classification problem. To maximize the performance, we used similar image removal and data augmentation [11] techniques. Our model was able to successfully confirm the pill confirmation with a high accuracy. This study can be used by medical professionals in virtually monitoring the patients and confirming whether the patients are taking medicines as prescribed and can take actions in case of non-compliance.

Future Work

In future, we can work on behavioral recognition of the patients and using Artificial intelligence, we can predict the behaviors of patients that could be really helpful in allocating proper resources to these patients in future.

Acknowledgments

The author(s) of this paper would like to thank the Radio Frequency and Auto Identification (RAID) Lab at the University of Texas - Arlington for supporting this project. The images taken for this study were provided by the research assistants working in the RAID Lab.

REFERENCES

[1] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks", 2019.

[2] J. Reisinho, M. Coimbra and F. Renna, "Deep Convolutional Neural Network Ensembles For Multi-Classification of Skin Lesions From Dermoscopic and Clinical Images," *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Montreal, QC, Canada,

2020, pp. 1940-1943, doi: 10.1109/EMBC44109.2020.9176411.

[3] M. A. Rahman, M. S. Hossain, N. A. Alrajeh and N. Guizani, "B5G and Explainable Deep Learning Assisted Healthcare Vertical at the Edge: COVID-I9 Perspective," in *IEEE Network*, vol. 34, no. 4, pp. 98-105, July/August 2020, doi: 10.1109/MNET.011.2000353.

[4] T. Carneiro, R. V. Medeiros Da NóBrega, T. Nepomuceno, G. Bian, V. H. C. De Albuquerque and P. P. R. Filho, "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications," in IEEE Access, vol. 6, pp. 61677-61685, 2018, doi: 10.1109/ACCESS.2018.2874767.

[5] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, 2015, pp. 730-734, doi: 10.1109/ACPR.2015.7486599.

[6] T. Treebupachatsakul and S. Poomrittigul, "Bacteria Classification using Image Processing and Deep learning," 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), JeJu, Korea (South), 2019, pp. 1-3, doi: 10.1109/ITC-CSCC.2019.8793320.

[7] P. He, "Study on Epidemic Prevention and Control Strategy of COVID -19 Based on Personnel Flow Prediction," 2020 International Conference on Urban Engineering and Management Science (ICUEMS), Zhuhai, China, 2020, pp. 688-691, doi: 10.1109/ICUEMS50872.2020.00150.

[8] F. Sabahi, M. O. Ahmad and M. N. S. Swamy, "Content-based Image Retrieval using Perceptual Image Hashing and Hopfield Neural Network," 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), Windsor, ON, Canada, 2018, pp. 352-355, doi: 10.1109/MWSCAS.2018.8623902.

[9] J. Ding, X. Li and V. N. Gudivada, "Augmentation and evaluation of training data for deep learning," 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, 2017, pp. 2603-2611, doi: 10.1109/BigData.2017.8258220.

[10] H. Ucuzal, A. K. Arslan and C. Çolak, "Deep learning based-classification of dementia in magnetic resonance imaging scans," 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 2019, pp. 1-6, doi: 10.1109/IDAP.2019.8875961.

[11] H. Li, J. Li, X. Guan, B. Liang, Y. Lai and X. Luo, "Research on Overfitting of Deep Learning," 2019 15th International Conference on Computational Intelligence and Security (CIS), Macao, Macao, 2019, pp. 78-81, doi: 10.1109/CIS.2019.00025.

[12] J. Kolluri, V. K. Kotte, M. S. B. Phridviraj and S. Razia, "Reducing Overfitting Problem in Machine Learning Using Novel L1/4 Regularization Method," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), Tirunelveli, India, 2020, pp. 934-938, doi: 10.1109/ICOEI48184.2020.9142992.

[13] S. V. Georgakopoulos and V. P. Plagianakos, "Efficient Learning Rate Adaptation for Convolutional Neural Network Training," 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 2019, pp. 1-8, doi: 10.1109/IJCNN.2019.8852033.

[14] Y. Ho and S. Wookey, "The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling," in IEEE Access, vol. 8, pp. 4806-4813, 2020, doi: 10.1109/ACCESS.2019.2962617.

[15] "Detect and Remove Duplicate Images from a Dataset for Deep Learning." *PyImageSearch*, 17 Apr. 2021, www.pyimagesearch.com/2020/04/20/detect-and-remove-duplicate-images-from-a-dataset-for-deep-learning/.

[16] Tobias, et al. "Image Hashing with OpenCV and Python." *PyImageSearch*, 17 Apr. 2021, www.pyimagesearch.com/2017/11/27/image-hashing-opencv-python/.

[17] "Detect and Remove Duplicate Images from a Dataset for Deep Learning." *PyImageSearch*, 17 Apr. 2021, www.pyimagesearch.com/2020/04/20/detect-and-remove-duplicate-images-from-a-dataset-for-deep-learning/.