A CONVOLUTIONAL NEURAL NETWORK (CNN) APPROACH TO DETECT FACE USING TENSORFLOW AND KERAS

Reny Jose
Marian college kuttikkanam (autonomous), Peermade, Idukki, Kerala

Abstract : Face recognition is used in a variety of aspects in the modern world. Face detection means to identify the face from a digital image. The deep neural network is considered a powerful tool as it can handle huge amounts of data .conventional neural network is one most popular tool to detect face detection. In this paper, a deep convolutional neural network (CNN) to extract features from input images. Keras is used for implementing CNN also D'lib and OpenCV for aligning faces on input images. Face recognition performance is evaluated using a custom dataset.

***IndexTerms* - Machine learning. Deep learning. Convolutional neural network OpenCV, Face Recognition.**

I. INTRODUCTION

The Face recognition is a hot research field in computer vision. Face Recognition begins with extracting the coordinates of features such as the width of the mouth, width of eyes, pupil, and compare the result with the measurements stored in the database and return the closest record (facial metrics). Nowadays, there are a lot of face recognition techniques and algorithms found and developed around the world. Facial recognition becomes an interesting research topic. It is proven by the number of published papers related to facial recognition including facial feature extraction, facial algorithm improvements, and facial recognition implementations.

Convolutional Neural Network (CNN) based on TensorFlow, an open-source deep learning framework, is proposed for face recognition. Convolutional Neural Network (CNN) also known as ConvNet architectures use to make the explicit assumption as the inputs are images, which allows the user to encode some properties into the architecture. These then make the forward function more efficient to implement and reduce the number of parameters in the network. The weights which should be loaded to the model using the Dlib's landmarks detector, and finally the custom dataset of images that should be loaded to the model.

I am using a pre-trained model which is the NN4.smallV2 model which detects the essential features from the input image. I have used the pre-trained model because many models are trained for different purpose as my project is in face recognition many of the intelligent people already created a model that could fetch essential information from the images. So to that case, we don't want to build a model for our purpose because it is a more time-consuming process. As a result, I have used a pertained model for recognizing individuals.

Using TensorFlow, which is an open-source artificial intelligence library developed by Google, we have studied and compared the effects of multiple activation functions on classification results. The functions used are Rectified Linear Unit (ReLu), Hyperbolic Tangent (tanH), Exponential Linear Unit (eLu), sigmoid, soft plus and softsign. It has a comprehensive, flexible ecosystem of tools, libraries which helps for easy model building, robust and powerful experimentation for research. OpenCV is an open-source library for image and video analysis, originally introduced more than a decade ago by Intel. Since then, several programmers have contributed to the most recent library developments. Keras is an open-source neural network library written in python and which runs on top of TensorFlow.

II. LITERATURE REVIEW

A literature review is a scholarly paper, which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic. Based on [1, 4, 5, 8] we used Tensor flow, one of the most popular deep learning libraries to classify the MNIST dataset, which is frequently used in data analysis studies. The functions used are Rectified Linear Unit (ReLu), Hyperbolic Tangent (tanH), Exponential Linear Unit (eLu), sigmoid, softplus and softsign. Convolutional Neural Network (CNN) and SoftMax classifier are used as deep learning artificial neural networks. The results show that the most accurate classification rate is obtained using the ReLu activation function

Face recognition [2 3 9] is has a high practical value for the detection and recognition of specific sensitive characters. The research found that in traditional hand-crafted features, there are uncontrolled environments such as pose, facial expression, illumination and occlusion influencing the accuracy of recognition and it has poor performance, so the deep learning method is adopted. Based on face detection, a Convolutional Neural Network (CNN) based on Tensor Flow, [2] an open-source deep learning framework, is proposed for face recognition. Experimental results show that the proposed method has better recognition accuracy and higher robustness in a complex environment. OpenCV (Open Source Computer Vision) [3] is an open-source library for image and video analysis, originally introduced more than a decade ago by Intel. Since then, several programmers have contributed to the most recent library developments

III. CONVOLUTIONAL NEURAL NETWORK (CNN)

Convolutional neural networks is a deep learning architecture that is used for face recognition. The Convolutional Neural Network (CNN) also known as ConvNet architectures is composed of a group of layers based on their functionality [2]. The network is able to capture the spatial and temporal dependencies in an array through the application of appropriate filters. It performs better filtering because of the reduction in the number of parameters involved and the reusability of weights.

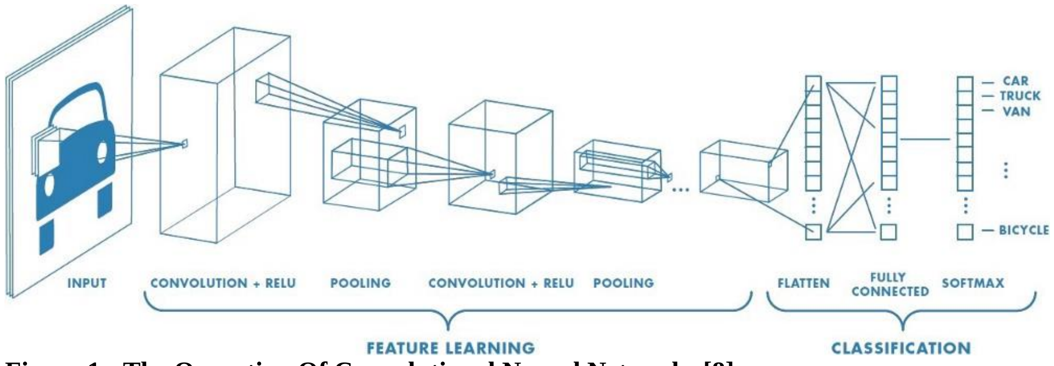

Figure 1 : The Operation Of Convolutional Neural Networks [8]

Main types of layers to build ConvNet architectures: Convolutional Layer, Pooling Layer, and Fully-Connected Layer. We will stack these layers to form a full ConvNet architecture [13].

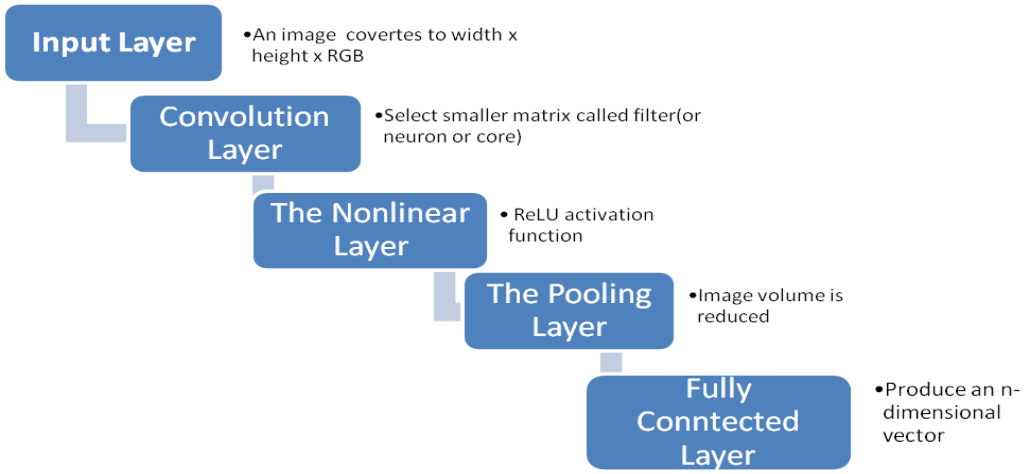

Figure :2 The Structure Of Convolutional Neural Networks

The elements used to perform convolutional operations at first part of the layer is known as the kernel or Filter or K. The first layer is used for capture low-level features such as edge, color gradient orientation, etc. by adding layers it can able to capture high-level features and as a result the neural network can understand the images in the data set. We use valid padding if the neural network generates a reduced dimensionality and if the dimensionality increases or remains the same we used the same padding.

The pooling layer is responsible for reducing the spatial size of convolved features. This helps to reduce the computational power to process the data two types of pooling are used one is max pooling which returns the maximum value of the kernel and average pooling which returns the average value of the kernel.

The fully connected layer learners a possible nonlinear function in the space. The input now converted as a column vector which is applied to the feed-forward neural network and back propagation is applied at every iteration. It is classified using the SoftMax classification

IV. IMPLEMENTATION

Face recognition identifies persons on face images or video frames. In a nutshell, a face recognition system extracts features from an input face image and compares them to the features of labeled faces in a database. Comparison is based on a feature similarity metric and the label of the most similar database entry is used to label the input image. If the similarity value is below a certain threshold the input image is labeled as unknown. Comparing two face images to determine if they show the same person is known as face verification.

The facial extraction system works based on the following

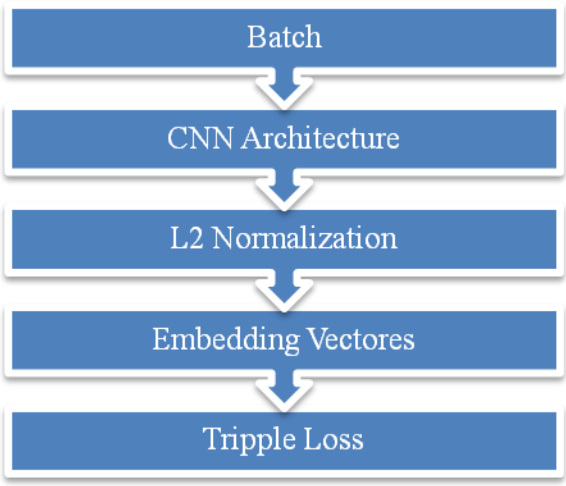

Figure 3: Workflow of Facial Extraction System

My work uses a deep convolutional neural network (CNN) to extract features from input images. Keras is used for implementing CNN also D'lib and OpenCV for aligning faces on input images. Face recognition performance is evaluated using a custom dataset in my project Face detection, Transformation, Cropping of faces takes place from the input images. This ensures that faces are aligned before feeding them into the CNN. This preprocessing step is very important for the performance of the neural network. By using CNN to extract 128-dimensional representations or embedding of faces from the aligned input images. In embedding space, Euclidean distance directly corresponds to a measure of face similarity. By using the Support Vector Machine (SVM) and K- Nearest Neighbors(KNN) classifiers we could predict the labels of the input images.

4.1 Environment setup

For running this project I have to create a virtual environment. Since I am using Anaconda, I have created the virtual environment inside anaconda. The virtual environment is created so that the packages that are listed in "requirement.txt" can be installed to that environment itself. This installation of packages specified in the text file can be installed with pip install -r requirements.txt in the terminal. Furthermore, we will need a local copy of Dlib's face landmarks data file for running face alignment.

4.2 Batch

Batch is considered to be the set of images that are taken into consideration for training the model. I took individuals pictures to train the model where this custom dataset is small and we could add as much as images of individuals if need in such a way. When reading the images by our model some images the model could not read the face from the image. So in such cases, we need to change the image.

4.3 CNN Architecture and Training

The CNN architecture I have used here is a variant of Inception architecture. More precisely it is a variant of the NN4 architecture which is identified as the NN4.small2 model and it is implemented in Keras. In this architecture, there is a fully-connected layer with 128 hidden units followed by an L2 Normalization Layer on top of the convolutional base. These two top layers are referred to like the embedding layer from which the 128-dimensional embedding vectors can be obtained. A Keras version of the NN4.small2 model can be created with create_model().

Model training aims to learn an embedding f(x) of image x such that the squared L2 distance between all faces of the same identity is small and the distance between a pair of faces from different identities is large. This can be achieved with a triplet loss **L** that is minimized when the distance between an Anchor image (x_i^a) and a Positive image (x_i^p) in embedding space is smaller than the distance between that anchor image and a negative image (x_i^n) by at least a margin α

$$L = \sum_{i=1}^{m} [\, || f(x_i^a) - f(x_i^p)) ||_2^2 \; - || f(x_i^a) - f(x_i^n)) ||_2^2 \; + \alpha]_+$$

$\alpha$is used in such a way that the distance of anchor-positive pairs and anchor-negative pairs should be far away from each other. Using the triplet_generator() function we could calculate the triplet loss.

Since we are using a pre-trained model we should provide the weights to our model and this can be availed by loading those weights to our model using a function load_weight() which is available within the pertained model.

4.4 Custom Dataset

To demonstrate face recognition on a custom dataset, a small subset of college mates is used. It consists of 100 face images of 10 Identities. The metadata for each image (file and identity name) are loaded into memory for later processing.

4.5 Face Alignment

The NN4.small2.v1 model was trained with aligned face images, therefore, the face images from the custom dataset must be aligned too. Here, we use D'lib for face detection and OpenCV for image transformation and cropping to produce aligned 96x96 RGB face images

The CNN Architecture used here NN4.SmallV2 which comes under the inception architecture. Usually, when designing a layer in a convolutional network, we may have to Choose 1 x 3, 3 x 3, 5 x 5 or a Pooling Layer. But for the case of inception architecture, all these are implemented together to make a layer. Their 'same' convolution is used to keep the dimension the same as of the previous layer. Using the concept of inception architecture, NN4.SmallV2 architecture is developed.

There these architecture is followed by an L2 Normalization layer. Few things are needed as inputs, it includes the model which we are using for processing the images, Dlib's landmarks detector, the weights which should be loaded to the model which we are using and finally the custom dataset of images which should be loaded to the model.

The pre-trained model which is NN4.smallV2 model which detects the essential features from the input image

4.6 L2 Normalization

The method of least absolute deviations (L2) finds applications in many areas, due to its robustness compared to the least-squares method (L1). Least absolute deviations are robust in that it is resistant to outliers in the data. Since an L2-norm square the error (increasing by a lot if error > 1), the model will see a much larger error (e1 vs e2) than the L1-norm, in our case we have used L2 Norm which helps to minimize the error

4.7 Embedding Vectors

Embedding vectors can be calculated by feeding the aligned and scaled images into the pre-trained network. Using the obtained embedding vectors we could calculate the distance between anchor-positive pair and anchor-negative pair and the distance calculated would be like given in the plot below

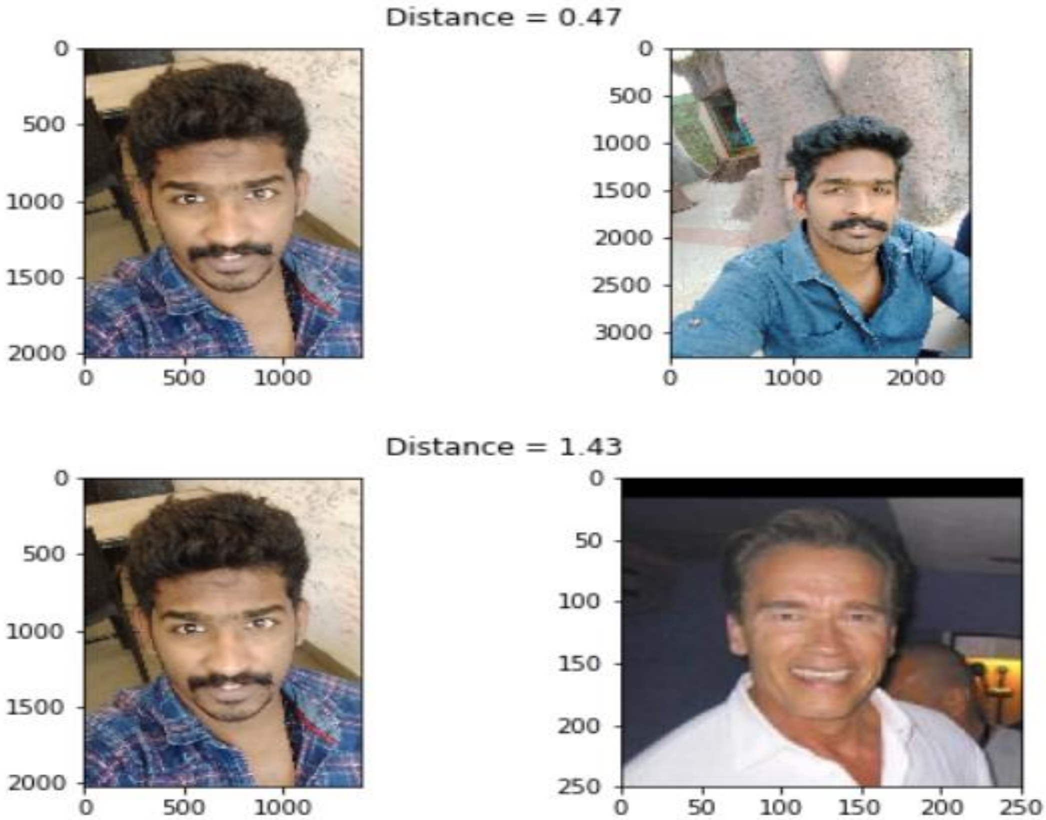

Figure 4: The distance calculated would be like given in the plot

4.8 Distance Threshold

To find the optimal value for the threshold, the face verification performance must be evaluated on a range of distance threshold values. At a given threshold, all possible embedding vector pairs are classified as either same identity or different identity and compared to the ground truth, since we are dealing with skewed classes I have used F1_score as the evaluation metrics.

4.9 Triplet Loss

We can train the network by taking an anchor image and comparing it with both a positive sample and a negative sample. The dissimilarity between the anchor image and positive image must low and the dissimilarity between the anchor image and the negative image must be high

V. RESULTS AND DISCUSSION

The ultimate result that we want is to detect individuals and recognize their identity as the output. At first, I have loaded my model which was implemented using inception architecture followed by an L2 Normalization layer. When two inception architecture was used then we can call that network as a siamese network. So why do we call as a Siamese network, this is because when we want to compare images to detect the person's identity we need two networks. This helps in finding the distance between the images that have passed through the network. This distance is calculated with the help of the Triplet Loss generator where it uses two pairs an anchor and a positive image and an anchor and a negative image. Whenever the distance which is calculated is greater than the threshold value then we could say that the two images that are feed into the network are not the same but if the distances between the pair of images are small then we could say that both the images are of the same individual. The calculations of distances are done if we convert the images into embedding vectors then only we could apply the concept of triplet loss generator. The output is generated with the help of the Support Vector Classifier (SVC). SVC allows predicting the identity of the individual. I have used KNN and SVC. But when it comes to the accuracy, SVC comes first. So I have used SVC to predict the identity of the individual. For training these classifiers I have used 50% of the dataset, for evaluation of the other 50%. The KNN classifier achieves an accuracy of 95% on the test set, the SVM classifier 97%. By using the SVM classifier I have got the face recognized accurately.

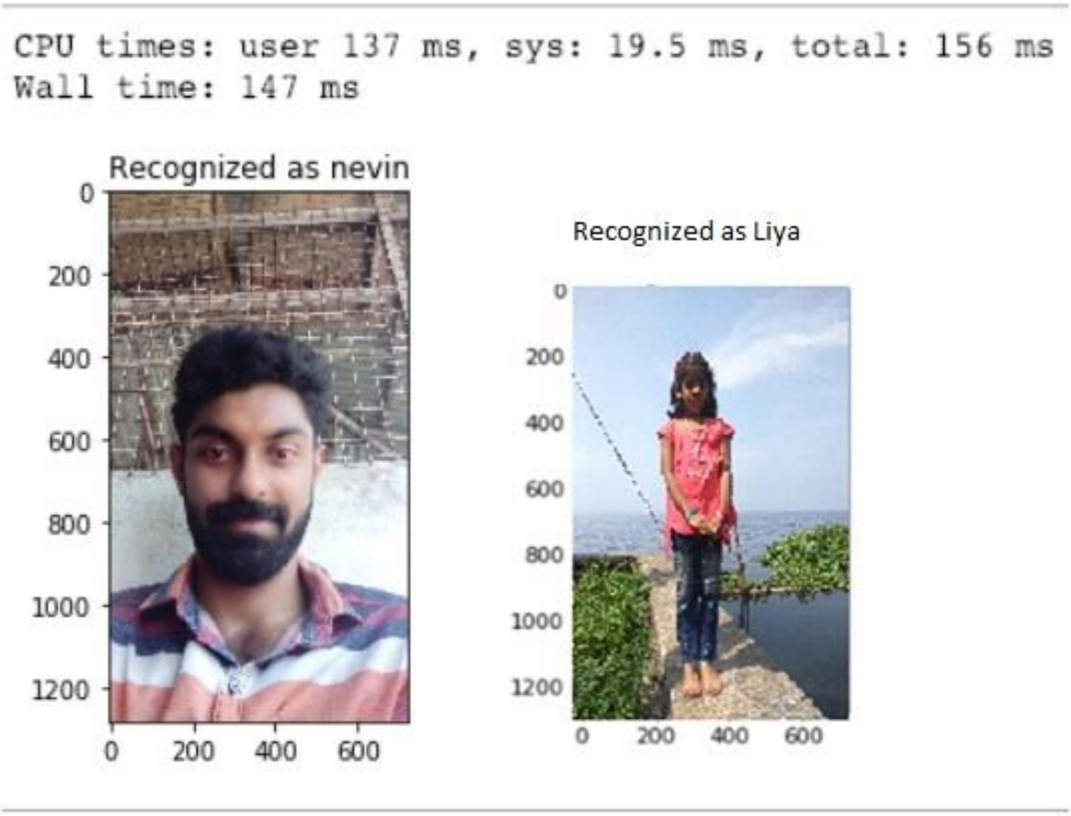

Figure 5: Recognized the person

VI. CONCLUSION AND FUTURE WORK

The convolutional network is very useful for facial recognition from images or videos. Face recognition is one of the newer developments of biometric identifiers that don't require as much time or intrude on the person its verifying. Face recognition is a highly effective biometric technology that holds a lot of potentials. Facial recognition is a very effective tool that can help law enforcers recognize criminals and software companies are leveraging the technology to help users access their technology. This technology can be further developed to be used in other avenues such as ATMs, accessing confidential files, or other sensitive materials. This can make other security measures such as passwords and keys obsolete. Another way that innovators are looking to implement facial recognition is within subways and other transportation outlets. They are looking to leverage this technology to use faces as credit cards to pay for your transportation fee.

REFERENCES

[1] Fatih Ertam; Galip Aydın Data classification with deep learning using Tensorflow- IEEE 2017 https://ieeexplore.ieee.org/document/8093521
[2] Liping Yuan; Zhiyi Qu; Yufeng Zhao; Hongshuai Zhang; Qing Nian " A convolutional neural network based on TensorFlow for face recognition"- - IEEE 2017 https://ieeexplore.ieee.org/document/8054070
[3] Ivan Culjak; David Abram; Tomislav Pribanic; Hrvoje Dzapo; Mario Cifrek "A brief introduction to OpenCV" 2012 International Conference. https://ieeexplore.ieee.org/document/6240859
[4] Dhakal, Parashar, Praveen Damacharla, Ahmad Y. Javaid, and Vijay Devabhaktuni. "A near real-time automatic speaker recognition architecture for voice-based user interface." Machine Learning and Knowledge Extraction 1, no. 1 (2019): 504-520.
[5] https://keras.io
[6] O. Abdel-Hamid, L. Deng, and D. Yu, "Exploring Convolutional Neural Network Structures and Optimization Techniques for Speech Recognition," no. August, pp. 3366–3370, 2013.
[7] http://www.deeplearningbook.org/contents/convnets.html
[8] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
[9] V. Dumoulin and F. Vision, "A guide to convolution arithmetic for deep learning," pp. 1–28, 2016.
[10] Zhao ZB, Fan XQ, Xu GZ et al (2017) Aggregating deep convolutional feature maps for insulator detection in infrared images. IEEE Access 5:21831–21839.
[11] Zuo GY, Ma L, Xu CF et al (2019) Insulator detection method based on cross-connected convolutional neural network. Autom Electr Power Syst 43(04):101–108
[12] Abdul-Malek Z (2017) Electrical and temperature correlation to monitor fault condition of Zno surge arrester. In: International conference on information technology, computer, and electrical engineering, IEEE, pp 182–186Google Scholar
[13] Ksenia Sorokina, Image Classification with Convolutional Neural Networks, 2018,https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8
[14] http://www.datamind.cz/cz/vam-na-miru/umela-inteligence-a-strojove-uceni-ai-machine-learning
[15] Junho Yim, Jeongwoo Ju, Heechul Jung, and Junmo KimImage Classification Using Convolutional NeuralNetworks With Multi-stage Feature Department of Electrical Engineering KAIST 291 Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea{creationi,veryju,heechul,junmo.kim}@kaist.ac.krhttps://sites.google.com/site/siitkaist
[16] Michael Grogan (MGCodesandStats), Image Recognition with Keras: Convolutional Neural Networks,Oct 31, 2018 https://towardsdatascience.com/image-recognition-with-keras-convolutional-neural-networks-e2af10a10114
[17] https://en.wikipedia.org/wiki/Convolutional_neural_network
[18] I. Kokkinos, E. C. Paris, and G. Group, "Introduction to Deep Learning Convolutional networks, Dropout, Maxout 1," pp. 1–70.
[19] https://www.lynda.com/Google-TensorFlow-tutorials/Building-Deep-Learning-Applications-Keras-2-0/601801-2.html
[20] Y. Guo, Y. Liu, A. Oerlemans, S. Wu, and M. S. Lew, "Author ' s Accepted Manuscript Deep learning for visual understanding: A review To appear i: Neurocomputing," 2015.
[21] Cheng Siong Chin, JianTing Si, A. S. Clare, Maode Ma, Intelligent Image Recognition System for Marine Fouling Using Softmax Transfer Learning and Deep Convolutional Neural Networks 2017
[22] https://www.opendatascience.com/blog/an-intuitive xplanationfconvolutionalneuralnetworks/?utm_source=Open+Data+Science+Newsletter &utm_campaign=f4ea9cc60fEMAIL_CAMPAIGN_2016_12_21& _medium=email & tm_term=0_2ea92bb125-f4ea9cc60f-245860601.
[23] https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/
[24] Krichevsky, A., Sutskever, I., and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks.In Advances in neural information processing systems (pp. u97-1105).
[25] N. Kwak, "Introduction to Convolutional Neural Networks (CNNs)," 2016.
[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z.Huang, A. Karpathy, C. V Jan, J. Krause, and S. Ma, "ImageNet Large Scale Visual Recognition Challenge.".
[27] D. Stutz and L. Beyer, "Understanding Convolutional Neural Networks," 2014.
[28] K. Teilo, " An Introduction to Convolutional Neural Networks," no.NOVEMBER 2015, pp. 0–11, 2016.
[29] https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html
[30] R. E. Turner, "Lecture 14 : Convolutional neural networks for computer vision," 2014.
[31] J. Wu, "Introduction to Convolutional Neural Networks," pp. 1–28,2016.
[32] https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
[33] http://www.slideshare.net/hanneshapke/introduction-to-convolutionalneural-networks.
[34] MEENA, D. and R. SHARAN. An approach to face detection and recognition. In: International Conference on Recent Advances and Innovations in Engineering (ICRAIE). Jaipur: IEEE, 2016, pp. 1–6. ISBN 978-1-5090-2807-8.DOI: 10.1109/ICRAIE.2016.7939462.
[35] REKHA, E. and P. RAMAPRASAD. An efficient automated attendance management system based on Eigen Face recognition. In: 7th International Conference on Cloud Computing, Data Science & Engineering – Confluence. Noida: IEEE, 2017, pp. 605–608. ISBN 978-1-5090-3519- 9. DOI: 10.1109/CONFLUENCE.2017.7943223.
[36] https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
[37] http://www.slideshare.net/hanneshapke/introduction-to-convolutionalneural-networks
[38] ABUROMMAN, A. A., and M. B. I. REAZ. Ensemble SVM classifiers based on PCA and LDA for IDS. In: International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES). Putrajaya: IEEE, 2016, pp. 95–99. ISBN 978-1-5090-2889-4. DOI: 0.1109/ICAEES.2016.7888016.

[39] OLIVARES-MERCADO, J., K. TOSCANOMEDINA, G. SANCHEZ-PEREZ, H. PEREZMEANA, and M. NAKANO-MIYATAKE. Face recognition system for smartphone based on LBP. In: 5th International Workshop on Biometrics and Forensics (IWBF). Coventry: IEEE, 2017, pp. 1–6. ISBN 978-1-5090-5791-7. DOI: 10.1109/IWBF.2017.7935111.

[40] KAMENCAY, P., T. TRNOVSZKY, M. BENCO, R. HUDEC, P. SYKORA, and A. SATNIK. Accurate wild animal recognition using PCA, LDA, and LBPH. In: ELEKTRO. Strbske Pleso: IEEE, 2016, pp. 62–67. ISBN 978-1-4673-8698-2. DOI: 10.1109/ELEKTRO.2016.7512036.

[41] AMEUR, B., S. MASMOUDI, A. G. DERBEL and A. BEN HAMIDA. Fusing Gabor and LBP feature sets for KNN and SRC-based face recognition. In: International Conference n Advanced Technologies for Signal and Image Processing (ATSIP). Monastir: IEEE, 2016, pp. 453– 458. ISBN 978-1-4673-8526-8. DOI: 10.1109/ATSIP. 2016.7523134.

[42] Esteva A, Kuprel B, Novoa RA et al (2017) Dermatologist-level classification of skin cancer with deep neural networks. Nature 542: 115–118

[43] Yasaka K, Akai H, Abe O, Kiryu S (2018) Deep learning with convolutional neural network for differentiation of liver masses at dynamic contrast-enhanced CT: a preliminary study. Radiology 286:887–896

[44] Liu F, Jang H, Kijowski R, Bradshaw T, McMillan AB (2018) Deep learning MR imaging-based attenuation correction for PET/MRimaging. Radiology 286:676–684

[45] Chen MC, Ball RL, Yang L et al (2018) Deep learning to classify radiology free-text reports. Radiology 286:845–852.