International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 08 | August - 2022 Impact Factor: 7.185 ISSN: 2582-3930

A Real-time Face Expression Detection using Convolutional Neural Networks

Prathibha G[1], H S Harsha[2], Manoj H S[3], Nisha H D[4], Chandana E[5]

[1]Assistant Professor, Department Computer Science & Engineering, Navkis College of Engineering, Hassan

[2,3,4,5]Department Computer Science & Engineering, Navkis College of Engineering, Hassan

---

Abstract - In this paper our group proposes and designs a featherlight convolutional neural network(CNN) for detecting facial feelings in real- time and in bulk to achieve a better bracket effect. We corroborate whether our model is effective by creating a real- time vision system. This system employment-task protruded convolutional networks(MTCNN) to complete face discovery and transmit the attained face coordinates to the facial feelings bracket model we designed originally. also it accomplishes the task of emotion bracket. Multi-task protruded convolutional networks have a waterfall discovery point, one of which can be used alone, thereby reducing the occupation of memory coffers. Our expression bracket model employs Global Average Pooling to replace the completely connected subcaste in the traditional deep intricacy neural network model. Each channel of the point chart is associated with the corresponding order, barring the black box characteristics of the completely connected subcaste to a certain extent.

Key Words: *Face Emotions, Convolutional Neural Networks, Biometrics, Gabor Filter*

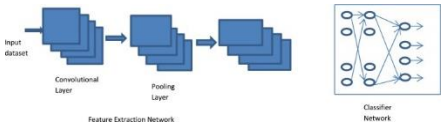
1.INTRODUCTION

Facial expression recognition software is a technology which uses biometric labels to descry feelings in mortal faces. More precisely, this technology is a sentiment analysis tool and is suitable to automatically descry the six introductory or universal expressions happiness, sadness, wrathfulness, surprise, fear, and nausea. Facial expression recognition or computer- grounded facial expression recognition system is important because of its capability to mimic mortal coding chops. Facial expressions and other gestures convey verbal communication cues that play an important part in interpersonal relations. These cues complement speech by helping the listener to interpret the intended meaning of spoken words. thus, facial expression recognition, because it excerpts and analyses information from an image or videotape feed, it's suitable to deliver undressed, unprejudiced emotional responses as data. Facial expression recognition system is a computer- grounded technology and thus, it uses algorithms to presently descry faces, law facial expressions, and fete emotional countries. It does this by analysing faces in images or videotape through computer powered cameras bedded in laptops, mobile phones, and digital signage systems, or cameras that are mounted onto computer defenses. For businesses, since facial expression recognition software delivers raw emotional responses, it can give precious information about the sentiment of a target followership towards a marketing communication, product or brand. It's the most ideal way to assess the effectiveness of any business content. Companies have traditionally done request exploration by conducting checks to find out about what consumers want and need. This system still,

assumes that the preferences stated are correct and reflect unborn conduct. But this isn't always the case. Another popular approach in request exploration is to employ behavioural styles where stoner's responses are observed, while interacting with a brand or a product. Although effective, similar ways can snappily come veritably labour ferocious as the sample size increases. In similar circumstances, facial expression recognition technology can save the day by allowing companies to conduct request exploration and measure moment- by-moment facial expressions of feelings automatically, making it easy total the results. Facial expression recognition can also be used in the videotape game testing phase. In this phase, generally a focus group of druggies is asked to play a game for a given quantum of time and their gesture and feelings are covered. By using facial expression recognition, game inventors can gain perceptivity and draw conclusions about the feelings endured during game play and incorporate that feedback in the timber of the final product.

2. Body of Paper

Facial expression recognition computer technology can obtain the emotional information of the



person through the expression of the person to judge the state and intention of the person. It's of great significance in mortal-computer commerce, safe driving, and intelligent advertising systems. It contains a series of images with the same expression ranging from calm to violent. We can prize neutral expression images from it. More precisely, this technology is a sentiment analysis tool and is suitable to automatically descry the six introductory expressions. General expression recognition styles include image reclaiming, facial point birth, and expression recognition. The reclaiming stage of image expression recognition performs face discovery to gain facial region images. The recognition of expressions in low- pixel facial images also requires image improvement or image super resolution during reclaiming. Image improvement is to enhance the being information of the image by changing the distribution of pixels, and image super resolution are to restore some missing pixel information by adding pixels.

Convolutional Neural Network

The name of similar networks is attained by applying a complication driver which is useful for working complex operations. The true fact is that CNNs give automatic point birth, which is the primary advantage. The specified input data is originally encouraged to a point birth network, and also the attendant uprooted features are encouraged to a classifier network. The software for emotion discovery undergoes training to insure that labors are correct and applicable. Understanding the inputs and labors is essential for algorithms. generally, speeches are transcribed into textbooks to be suitable to assay them, but this system would not be applicable in emotion recognition. colorful exploration is still ongoing to explore the operation of speech rather of transcribed textbooks for emotion recognition operations. Document- Term Matrix or DTM is the usual data structure used for textbooks. It's a matrix where records of the frequence of words in a document can be set up. still, it isn't applicable for determining feelings since it uses individual words.

Gabor Filter

A Gabor sludge, deduced from Gabor abecedarian functions(GEF), is a direct sludge used for a multitude of image processing operations for texture analysis, edge discovery, point birth, etc. As a band-pass sludge, the Gabor sludge enables the birth of patterns at the specified certain frequence and exposure of the signal. thus, this performing property of transubstantiating texture differences into sensible sludge- affair discontinuities at texture boundary has established itself to mimic the functionality of the visual cortex.

Original double Convolution Networks

In they've proposed original double complication(LBC), an effective volition to convolutional layers in standard convolutional neural networks(CNN). The design principles of LBC are motivated by original double patterns(LBP). The LBC subcaste comprises of a set of fixed meager pre-defined double convolutional pollutants that aren't streamlined during the training process, anon-linear activation function and a set of learnable direct weights.

METHODOLOGY

Modules

- Dataset
- Importing the necessary libraries
- Retrieving the images
- Splitting the dataset
- Building the model
- Apply the model and plot the graphs for accuracy and loss
- Accuracy on test set
- Saving the Trained Model

Dataset

In the first module, we developed the system to get the input dataset for the training and testing purpose. We have taken the dataset from Facial Expression Detection The dataset consists of 51137 Facial Expression images. The images are processed in such a way that the faces are almost centered and each face occupies about the same amount of space in each image. Each image has to be categorized into one of the seven classes that express different facial emotions. These facial emotions have been categorized as: 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, and 6=Neutral. Figure 1 depicts one example for each facial expression category. In addition to the image class number (a number between 0 and 6), the given images are divided into three different sets which are training, validation, and test sets. There are about 29,000 training images, 4,000 validation images, and 4,000 images for testing. After reading the raw pixel data, we normalized them by subtracting the mean of the training images from each image including those in the validation and test sets. For the purpose of data augmentation, we produced mirrored images by flipping images in the training set horizontally.

Importing the necessary libraries

We will be using Python language for this. First, we will import the necessary libraries such as keras for building the main model, sklearn for splitting the training and test data, PIL for converting the images into array of numbers and other libraries such as pandas, NumPy, matplotlib and TensorFlow.

Retrieving the images

We will retrieve the images and their labels. Then resize the images to (224,224) as all images should have same size for recognition. Then convert the images into NumPy array.

Splitting the dataset

Split the dataset into train and test. 80% train data and 20% test data.

Convolutional Neural Networks

The objectives behind the first module of the course 4 are:

- To understand the convolution operation
- To understand the pooling operation
- Remembering the vocabulary used in convolutional neural networks.
- Building a convolutional neural network for multi-class classification in images

Computer Vision

Some of the computer vision problems which we will be solving in this article are:

1. Image classification
2. Object detection
3. Neural style transfer

One major problem with computer vision problems is that the input data can get really big. Suppose an image is of the size 68 X 68 X 3. The input feature dimension then becomes 12,288. This will be even bigger if we have larger images (say, of size 720 X 720 X 3). Now, if we pass such a big input to a neural network, the number of parameters will swell up to a HUGE number (depending on the number of hidden layers and hidden units). This will result in more computational and memory requirements – not something most of us can deal with.

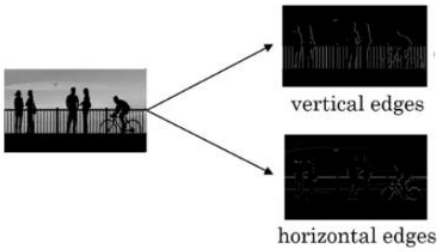
Edge Detection Example

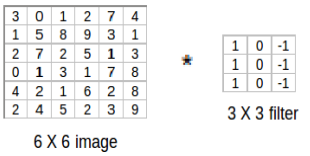
In the previous article, we saw that the early layers of a neural network detect edges from an image. Deeper layers might be able to detect the cause of the objects and even more deeper layers might detect the cause of complete objects (like a person's face).

In this section, we will focus on how the edges can be detected from an image. Suppose we are given the below image:

Fig 5.1: Edge Detection

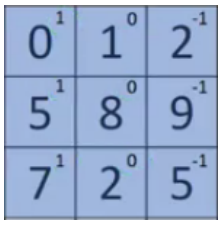
As you can see, there are many vertical and horizontal edges in the image. The first thing to do is to detect these edges:





After the convolution, we will get a 4 X 4 image. The first element of the 4 X 4 matrix will be calculated as:

So, we take the first 3 X 3 matrix from the 6 X 6 image and multiply it with the filter. Now, the first element of the 4 X 4 output will be the sum of the element-wise product of these values, i.e. 3*1 + 0 + 1*-1 + 1*1 + 5*0 + 8*-1 + 2*1 + 7*0 + 2*-1 = -5. To calculate the second element of the 4 X 4 output, we will shift our filter one step towards the right and again get the sum of the element-wise product:



Similarly, we will convolve over the entire image and get a 4 X 4 output:

-5	-4	0	8
-10	-2	2	3
0	-2	-4	-7
-3	-2	-3	-16

So, convolving a 6 X 6 input with a 3 X 3 filter gave us an output of 4 X 4. Consider

More Edge Detection

The type of filter that we choose helps to detect the vertical or horizontal edges. We can use the following filters to detect different edges:

1	0	-1
1	0	-1
1	0	-1

Vertical

1	1	1
0	0	0
-1	-1	-1

Horizontal

Some of the commonly used filters are:

1	0	-1
2	0	-2
1	0	-1

Sobel filter

3	0	-3
10	0	-10
3	0	-3

Scharr filter

The Sobel filter puts a little bit more weight on the central pixels. Instead of using these filters, we can create our own as well and treat them as a parameter which the model will learn using backpropagation.

Padding

We've seen that convolving an input of 6 X 6 dimension with a 3 X 3 sludge results in 4 X 4 affair. We can generalize it and say that if the input is n X n and the sludge size is f X f, also the affair size will be(n- f 1) X(n- f 1) • Input n X n • Sludge size f X f • Affair(n- f 1) X(n- f 1) There are primarily two disadvantages now 1. Every time we apply a convolutional operation, the size of the image shrinks. Pixels present in the corner of the image are used only a several numbers of times during complicacy as compared to the central pixels. Hence, we don't rivet too important on the corners since that can lead to information loss To overcome these issues, we can pad the image with an fresh border, i.e., we add one pixel each around the edges. This means that the input will be an 8 X 8 matrix(rather of a 6 X 6 matrix). Applying intricacy of 3 X 3 on it'll behave in a 6 X 6 matrix which is the original shape of the image. This is where padding comes to the fore • Input n X n • Padding p • adulterant size f X f • yield (n 2p- f 1) X(n 2p- f 1) There are two common choices for padding 1. Valid It means no padding. However, the product will be (n- f 1) X (n- f 1) If we're using valid padding.2. Same Then, we apply padding so that the affair size is the same as the input size, i.e., n 2p- f 1 = n So, p = (f- 1)/ 2 We now know how to use padded complication. This way we do n't lose a lot of information and the image doesn't shrink moreover. Next, we will look at how to apply strided complications.

Strided complications

Suppose we choose a stride of 2. So, while convoluting through the image, we will take two way – both in the vertical and perpendicular directions independently. The confines for stride s will be • Input n X n • Padding p • Stride s • Sludge size f X f •

Affair ((n 2p- f)/ s 1) X((n 2p- f)/ s 1) Stride helps to reduce the size of the image, a particularly useful point.

complications Over Volume Suppose, rather of a 2- D image, we've a 3- D input image of shape 6 X 6 X 3. How will we apply complication on this image? We'll use a 3 X 3 X 3 sludge rather of a 3 X 3 sludge. Let's look at an illustration • Input 6 X 6 X 3 • Sludge 3 X 3 X 3 The confines above represent the height, range and channels in the input and sludge. Keep in mind that the number of channels in the input and sludge should be same. This will affect in an affair of 4 X 4. Let's understand it visually Since there are three channels in the input, the sludge will accordingly also have three channels. After complication, the affair shape is a 4 X 4 matrix. So, the first element of the affair is the sum of the element-wise product of the first 27 values from the input(9 values from each channel) and the 27 values from the sludge. After that we convolve over the entire image. rather of using just a single sludge, we can use multiple pollutants as well. How do we do that? Let's say the first sludge will descry perpendicular edges and the alternate sludge will descry vertical edges from theimage.However, the affair dimension will change, If we use multiple pollutants. So, rather of having a 4 X 4 affair as in the below illustration, we'd have a 4 X 4 X 2 affair(if we've used 2 pollutants) Generalized dimensions can be given as:

- Input: n X n X n_c
- Filter: f X f X n_c
- Padding: p
- Stride: s
- Output: [(n+2p-f)/s+1] X [(n+2p-f)/s+1] X n_c'

Here, n_c is the number of channels in the input and filter, while n_c' is the number of filters.

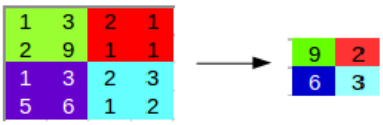
One Subcaste of a Convolutional Network Once we get an affair after convolving over the entire image using a sludge, we add a bias term to those labors and eventually apply an activation function to induce activations. This is one subcaste of a convolutional network. Recall that the equation for one forward pass is given by z(1) = w(1) * a(0) b(1) a(1) = g(z(1)) In our case, input(6 X 6 X 3) is a(0) and pollutants(3 X 3 X 3) are the weights w(1). These activations from subcaste 1 act as the input for subcaste 2, and so on. easily, the number of parameters in case of convolutional neural networks is independent of the size of the image. It basically depends on the sludge size. Suppose we've 10 pollutants, each of shape 3 X 3 X 3. • Number of parameters for each sludge = 3 * 3 * 3 = 27 • There will be a bias term for each sludge, so total parameters per sludge = 28 • As there are 10 pollutants, the total parameters for that subcaste = 28 * 10 = 280 • f(l) = sludge size • p(l) = padding • s(l) = stride • n(c)(l) = number of pollutants

Simple Convolutional Network Example We 'll take effects up a notch now. Let's look at how a complication neural network with convolutional and pooling subcaste workshop. Suppose we've an input of shape 32 X 32 X 3 We take an input image(size = 39 X 39 X 3 in our case), convolve it with 10 pollutants of size 3 X 3, and take the stride as 1 and no padding. This will give us an affair of 37 X 37 X 10. We convolve

this affair further and get an affair of 7 X 7 X 40 as shown over. Eventually, we take all these figures(7 X 7 X 40 = 1960), untwine them into a large vector, and pass them to a classifier that will make prognostications. This is a exemplification of how a convolutional network workshop. There are a number of hyperparameters that we can tweak while erecting a convolutional network. These include the number of pollutants, size of pollutants, stride to be used, padding, etc. We'll look at each of these in detail latterly in this composition. Just keep in mind that as we go deeper into the network, the size of the image shrinks whereas the number of channels generally increases. The part of the ConvNet is to reduce the images into a form which is easier to reuse, without losing features which are critical for getting a good vaticination. This is important when we're to design an armature which isn't only good at learning features but also is scalable to massive datasets. The armature performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the complication of the image more. In a convolutional network(ConvNet), there are principally three types of layers . Convolution subcaste 2. Pooling subcaste 3. Completely connected subcaste Pooling Layers Pooling layers are generally used to reduce the size of the inputs and hence speed up the calculation. Consider a 4 X 4 matrix as shown below

1	3	2	1
2	9	1	1
1	3	2	3
5	6	1	2

Applying max pooling on this matrix will result in a 2 X 2 output:



For every consecutive 2 X 2 block, we take the max number. Here, we have applied a filter of size 2 and a stride of 2. These are the hyperparameters for the pooling layer. Apart from max pooling, we can also apply average pooling where, instead of taking the max of the numbers, we take their average. In summary, the hyperparameters for a pooling layer are:

1. Filter size
2. Stride
3. Max or average pooling

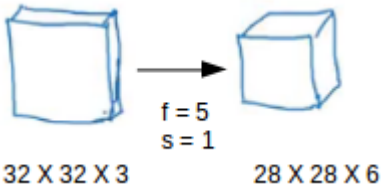
If the input of the pooling layer is n_h X n_w X n_c, then the output will be [{(n_h – f) / s + 1} X {(n_w – f) / s + 1} X n_c].

CNN Example

1. We ’ll take effects up a notch now. Let’s look at how a complication neural network with convolutional and pooling subcaste workshop. Suppose we've an input of shape 32 X 32 X 3 There are a combination of complication and pooling layers at the morning, a many completely connected layers at the end and eventually a SoftMax classifier to classify the input into colorful orders. There are a lot of hyperparameters in this network which we've to specify as well. Generally, we take the set of hyperparameters which have been used in proven exploration and they end up doing well. As seen in the below illustration, the height and range of the input shrinks as we go deeper into

the network(from 32 X 32 to 5 X 5) and the number of channels increases(from 3 to 10). There are primarily two major advantages of using convolutional layers over using just completely connected layers Parameter sharing

2. Sparsity of connections



If we would have used just the fully connected layer, the number of parameters would be 32*32*3*28*28*6, which is nearly equal to 14 million. Still, it'll be = (5 * 5 1) * 6(if there are 6 pollutants), which is equal to 156, If we see the number of parameters in case of a convolutional subcaste. Convolutional layers reduce the number of parameters and speed up the training of the model significantly. In complications, we partake the parameters while convolving through the input. The suspicion behind this is that a point sensor, which is helpful in one part of the image, is presumably also useful in another part of the image. So a single sludge is convolved over the entire input and hence the parameters are participated. The alternate advantage of complication is the sparsity of connections. For each subcaste, each affair value depends on a small number of inputs, rather of taking into account all the inputs. Erecting the model Apply the model and plot the graphs for delicacy and loss We'll collect the model and apply it using fit function. The batch size will be 10. also we will compass the graphs for delicacy and loss. We got average confirmation delicacy of87.6 and average training delicacy of98.3. Delicacy on test set We got an delicacy of96.7 on test set Saving the Trained Model Once you 're confident enough to take your trained and tested model into the product-ready terrain, the first step is to save it into a. h5 or. pkl train using a library like fix. We should have fix installed in your terrain. Next, let's import the module and leave the model into. pkl train

RESULTS AND DISCUSSION

To compare the performance of the shallow model with the deep model, we colluded the loss history and the attained delicacy in these models. numbers 3 and 4 parade the results. As seen in Figure 4, the deep network enabled us to increase the confirmation delicacy by18.46. As demonstrated in these numbers, the deep network results in advanced true prognostications for utmost of the markers. It's intriguing to see that both models performed well in prognosticating the happy marker, which implies that learning the features of a happy face is easier than other expressions. also, these matrices reveal which markers are likely to be confused by the trained networks. For illustration, we can see the correlation of angry marker with the fear and sad markers. There are lots of cases that their true marker is angry but the classifier has misclassified them as fear or sad. These miscalculations are harmonious with what we see when looking at images in Expression Shallow Model Deep Model Angry 41 53 nausea 32 70 Fear

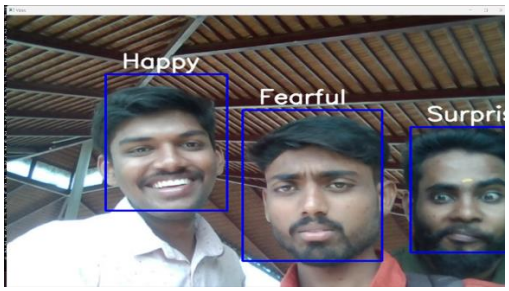


Fig 6.1: Face Expression Results

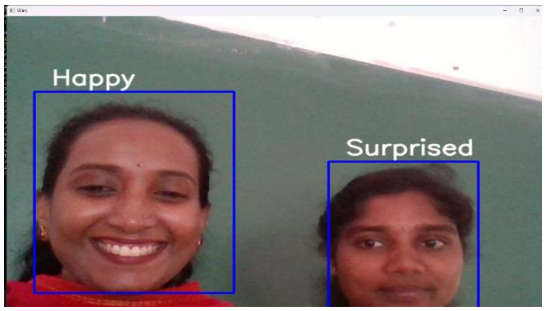


Fig 6.2: Face Expression Results obtained for Happy and Surprised Test Cases

54% 46% Happy 75% 80.5% Sad 32% 63% Surprise 67.5% 62.5% Neutral 39.9% 51.5% Table 3: The accuracy of each expression in the shallow and deep models. the dataset; indeed as a mortal, it can be delicate to fete whether an angry expression is actually sad or angry. This is due to the fact that people don't all express feelings in the same way. In addition to confusion matrices, we reckoned the delicacy of each model for every expression. **CONCLUSION**

We developed colourful CNNs for a facial expression recognition problem and estimated their performances using different post-processing and visualization ways. The results demonstrated that deep CNNs are able of learning facial characteristics and perfecting facial emotion discovery. Also, the mongrel point sets didn't help in perfecting the model delicacy, which means that the convolutional networks can naturally learn the crucial facial features by using only raw pixel data. Aiming at the expression recognition of low- pixel face images, the paper proposes an advanced CNN expression recognition system. The composition increases the nonlinearity of the network model by adding a convolutional subcaste. We can learn from excerpt image features in further layers and reflect image information.

REFERENCES

1. Saad ALBAWI , Tareq Abed MOHAMMED , Saad AL-ZAWI **,"Understanding of a Convolutional Neural Network"** 2017

2.Reagan L. Galvez, Argel A. Bandala, Elmer P. Dadios**, "Object Detection Using Convolutional Neural Networks",** October 2018

3.Tahmina Zebin, Patricia J. Scully, Niels Peek, Alexander J. Casson**, "Design and Implementation of a Convolutional neural network on an edge Computing**

smartphone for human Activity recognition"

4.Rahul Chauhan, Kamal Kumar Ghanshala, R.C Joshi **"Convolutional Neural Network(CNN) for Image Detection and Recognition"** 2018

5. Kyong Hwan Jin, Michael T. McCann, Member, IEEE, Emmanuel Froustey, and Michael Unser**, "Deep Convolutional Neural Network for Inverse Problems in Imaging",** 2017.

6. Balaji Balasubramanian, Rajeshwar Nadar, Pranshu Diwan, Anuradha Bhatia,**"Analysis of Facial Emotion Recognition"** 2016

7.Lei Gao, Lin Q, Ling Guan**, "Sparsity Preserving Multiple Canonical Correlation Analysis with Visual Emotion Recognition to Multi-Feature Fusion"**

8. Ketki R. Kulkarni , Sahebrao B. Bagal **, "Facial Expression Recognition"**

9. Khadija Lekdioui, Yassine Ruichek, Rochdi Messoussi, Youness Chaabi, Rajaouahni **"Facial Expression Recognition Using Face-Regions"**

10.Boris Knyazev, Roman Shvetsoy, Natalia Efremova **"Leveraging large face recognition data for emotion classification"** 2018