IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 6.252

Volume 2, Issue 2, December 2022

Image Colourization using Clustering Technique

Dr. S. Venkatakiran[1], R. Raghavi[2], J. Priyanka[3], K. Tharun[4], T. Perumal[5], C. Naveen[6], K. Sivapriya[7]

Assistant Professor, Department of Electronics and Communication Engineering[1]

UG Students, Department of Electronics and Communication Engineering[2,3,4,5,6]

Sri Venkatesa Perumal College of Engineering and Technology, Puttur, AP, India

Abstract: *Colorization of grayscale images has become a more researched area in the recent years, thanks to the advent of K-mean clustering networks. We attempt to apply this concept to colorization of real images obtained from video sequences. Previous similar research focused mainly colorization of natural images, while colorization of real is traditionally done by leveraging manual scribble methods. Our proposed method is a fully automated process. To implement it, we propose and compare two distinct K-mean clustering architectures trained under various loss functions. We aim to compare each variant based on results obtained as individual images.*

Keywords: Gray scale, Pixel, Automatic color image, K-mean clustering, Python, NumPy, OpenCV.

I. INTRODUCTION

An image is defined as a two-dimensional function, F(x, y), where x and y are spatial coordinates, and the amplitude of F at any pair of coordinates (x, y) is called the intensity of that image at that point. A digital image is a two-dimensional array of pixels. Each pixel has an intensity value (represented by a digital number) and a location address (referenced by its row and column numbers). When x, y, and amplitude values of F are finite, we call it a digital image. The images may be analog or digital. Aerial photographs are examples of analog images while satellite images acquired using electronic sensors are examples of digital images. A digital image is a two-dimensional array of pixels. Binary images can take one two value 0 and 1 or typically black and white. Binary images take only 1 binary digit to represent each pixel so it is also known as 1-bit image e.g.-optical character recognition (OCR). Gray-level images only contain gray level information they do not contain any color information. Available number of different gray levels is determined by the number of bits used for each pixel. For example: for 256 different gray level gray scale images should contain 8bits/pixel data.12 or 16bits/pixel data are used for the medical imaging and astronomy. A color image is an image which is represented by some color space. This color space is not dependent on only one value like greyscale image. Each pixel in color image is represented by more than one value and combined effect of these values gives appearance of a color. Multispectral images contain the information outside the normal human perceptual range. Information represented is not directly visible by human system so, these are not images in the usual sense.

However, by mapping the different spectral band to RGB components the information is represented in visual form. Multispectral images include the ultraviolet, infrared, X-ray, radar data and acoustic military target identification.

II. LITERATURE SURVEY

- Colorization basically involves assigning realistic colors to grey-scale image. K-mean clustering networks are specifically designed to deal with image data. Many authors have done promising work on this idea.
- Hsin-Ying Lee [2020] proposed the idea of automatic coloring of images, since they arevery different from natural images; they pose a difficulty as their colors depend on artist to artist. So, the data-set was specifically trained for images, about 100000 images, 70% of which were used in training and rest for validation. But unfortunately, the color uncertainty in is much higher than in natural images and evaluation is subjective and slow.
- Saeed Anwar, Muhammad Tahir [2020] proposed another similar approach, employing the use of Google's image classifier, Inception ResNet V2. The system model is divided into 4 parts, Encoder, Feature extractor, Fusion layer and Decoder. The system is able to produce acceptable outputs, given enough resources, CPU, Memory, and large data-set. This is mainly proof of concept implementation.
- Tsai-Ho Sun [2019] proposed a approach to mainly address the problem of coloring Chinese films from past

time. They used existing data-set with their data-set of Chinese images, fine-tuning the overall model. The network makes use of multi- scale convolution kernels by adversal counter colors, combining low and middle features extracted from VGG-16

- Kamyar Nazeri [2019] proposed a method to convert plain sketches into colorful images. It uses sketch inversion model and color prediction based on their edge detection in CIELab color space. This approachis able to handle hand drawn sketches including various geometric transformations. The limitation found was that, data-set is very limited but it works well for uncontrolled conditions.
- Geirhos et al [2019] has proposed a optimized solution by using huge data-set and singlefeed- forward pass in K-mean clustering. Their main focus on training part. They used human subjects to test the results and were able to fool 32% of them can have various number of neurons. The various attempts used various architectures.

In some papers, generally number of neurons is same as the dimension of the feature descriptor extracted from each pixel coordinates in a gray-scale image. For colorizing grey images, there are two prominent research works in this field.

III. EXISTING METHOD

The present existing method uses human assistance to work. This method, termed as Scribble method, divides image into definitive regions based on the boundaries accordingly. Each region is then separately applied selective color to these regions. Lot of advancements progressively improving the ease of applying colors by improving the methods and algorithms. These are generally procedural and require step by step progress. The non-iterative method combined with the adaptive edge extraction was proposed to reduce the running time of colorization optimization and color bleeding effects. Different with previous methods, Yatziv & Sapiro presented the idea of colour blending, which means that the chrominance value of a pixel is the result of contribution from given colors. To achieve a better visual effect when user strokes are sparse, Luan model proposed to consider both the neighboring pixels with similar intensity and remote pixels with similar textures. For the same purpose, Xu model, applied the probability distribution to find out the most confident stroke color for every pixel and thus decide the color of pixel. However, intensity-based colorization methods used in colorizing grayscale images may fail to colorize sketch because manga has no grayscale information. Qu-model designed a method that can propagate color through pattern-continuous regions as well as intensity continuous regions (Qu model). Later, Sykora and colleagues pointed out the limitations in Qu model and formulated their ideal painting tool by converting the properties to an energy minimization framework in order to colorize black-and-white cartoons more conveniently (Sýkora model). With the development of deeplearning, Zhang model incorporated U-Net structure into their colorization framework to reduce user's interventions and improve the visual effect of colorization. In case that number of pixels in color image cluster is less than number of pixels in grey image cluster, replication of values in feature vector distance is done starting from lowest values of feature vector distance. Reason for replication of lower values is chosen because of the fact that the higher values replication results in noisy pixels. This process results in equal number of pixels in matched pair of clusters. Next, chromatic values of Lab are copied from pixelsin color image cluster to pixels in grey image clusters.

Finally convert image back to RGB color space, since we converted color image to Lab color space for going through this processing. To simplify training and reduce memory requirements to tractable amounts, we down sample the input images to 224*224 resolution and output a 56*56 resolution colorization, which is scaled up to match the grayscale input dimensions.

3.1 Drawbacks of Existing System

The above details methods have several drawbacks and the dominant ones are detailed further.

- Region based coloring is highly time consuming.
- Requires lot of man power since each picture has to be colored separately and individually one at a time.
- Accuracy of the applied color is very low and prone to human error.
- Repetitive work since there is no permanent solution as the method is a step-by-step progression.

IV. PROPOSED METHOD

The present proposed system is colorization with K-mean clustering network with the development of deep learning, many studies of different areas incorporate deep learning networks into their frameworks. Colorization, as one of those areas, can be improved by the help of deep learning networks.

4.1 Methodologies

All methods described previously, with the exception of deep colorization, require some form of user assistance, which reduces their theoretical throughput for large scale colorization and can make them inconvenient to use, possibly resulting in having to resort to trial-and-error in order to obtain a satisfactory result. Given that their running times are generally in orders of minutes, that makes them not suitable for colorizing too many images. However, with the K-mean clustering boom described in Section some researchers started tackling this to a higher degree of success as a fully automated process by training K-mean clustering's on large datasets such as SUN or ImageNet. Currently, these methods are the state-of-the-art for natural image colorization. It is worth noting that even the application of K-mean clustering's to this task can be viewed as a form of automatic color or style transfer - with the references automatically pre- selected by the choice of the original K-mean clustering training set - using a complex method learned and realized by the network. However, by using training set which contains a large variety of semantically different scenes and commonly occurring objects (such as ImageNet), it is expected that the chosen color transfer should be the best matching one. Zhang model propose a plain K-mean clustering with 22 convolutional layers on a subset of the ImageNet dataset, employing a custom tailored multinomial cross entropy loss with class rebalancing based on prior color distribution obtained from the training set to predict a colour histogram for each output pixel to handle the multi-modal nature of the task. Similarly, Larsson model also predict a color histogram, however, they choose to use a 16-layer convolutional model attached to a fully connected hyper column layer to predict pixels' chromatic values, pre-trained on image classification task and fine-tuned for colorization. Rather than train densely and predict the colorization of the whole image in one pass, the K-mean clustering is trained on spatially sparse samples of grayscale patches of size equal to the receptive field of the network, predicting the color value of the central pixel.

Larsson model also explore the possibility of transferring a known ground truth color histogram (as a global descriptor) to improve the colorization. Iizuka model propose a network which combines two paths of computation, one to predict the global features of the target image and the other to specialize in local features. To achieve this, the global features are trained for image classification rather than colorization and are subsequently concatenated to the local features that are trained directly for colorization using L2 Euclidean loss function. This technique allows their model to gain a higher semantic understanding of the image. The training step first involves converting the image which is going to be sent to the model, to a resized format of 256*256. After this step we use the color image and using a function we convert it into greyscale form as is required by the learning model. Now, using NumPy, we translate the image into a matrix format. The tensor's present in the model will then be able to extract features of these images from the matrix format as obtained above. This information from the tensors is critical to the learning process of the model. This recognition of aspects of the image is part of the learning and depends on the learning rate of the model. As we use Keras framework, the learning rate is automated. Certain optimizers such as Adam and RMS prop are put to use that help to make adjustments to the learning rate. The method is quite straight forward and computationally efficient, it is very useful for us. Learning rate schedules seek to adjust the learning rate during training by reducing it according to a predefined schedule. Common learning rate schedules include time based decay, step decay and exponential decay. By experimenting with the range of learning rates, the accuracy of our model improves. After considerable amount of parameter tuning, our training process is done. We are now readyto put our training model to test. We managed to train our model on a diverse range of images to improve its knowledge base. Thus, for our testing process, we pass through random images that can represent any entity. To begin with our testing phase, we need a greyscale image which needs to be colorized by our model. We make sure we have converted/resized the image based on the constraints of the code we have written. The image is then captured by our model. It then gets translated by NumPy to the matrix form for the model to extract features of the image,producing very consistent colorizations Zhanget al, proposed to train the K-mean clustering with many images and thus applied this network to colorize new grayscale images. Specifically, Zhangetal implemented the U-Net structure (Ronneberger etal) to realize colorization, designing feature extraction part, global input part, fusion part and reconstruction part. However, during the process of training the K-mean clustering, it's very difficult to collect user input data. Toovercome this difficulty, they obtained the simulating user inputs by sampling several points on colorful images.

The other modules included in our code are Skimage and NumPy. Skimage is a collection of algorithms for image processing and computer vision. We have used this collection for performing functions and changes to the images we are importing to our code for both training as well as implementation, for example for saving the output images, resizing the images as well as converting the pixels in the images into Ubyte. We have also NumPy for the mathematical computations on the pixel values we have obtained for each of these images. Thismakes it easy for performing the necessary algorithms on a matrix.

From the above figure 4.1 this task, our goal is to take a grayscale image, a single channel of image data, and transform it into a standard RGB image, an image with three channels of data. We pose the problem as learning a mapping C such that R = C(G), where R ∈ R H×W×3 represents an image with RGB channels and G ∈ R H×W represents an image with grayscale values only. To learn this mapping, we use K-mean clustering based models. To simplify training and reduce memory requirements to tractable amounts, we down sample the input images to 224*224 resolution and output a 56*56resolution colorization, which is scaled up to match the grayscale input dimensions. Despite the radical decrease in resolution, it should be possible to learn a good looking colorization, as evident from the work of Zhang model, even if some finer detail is lost. The input's total number of pixels is 50176 and output's total number is 3136. To up sample the colorized result, we use a conventional spline interpolation algorithm.

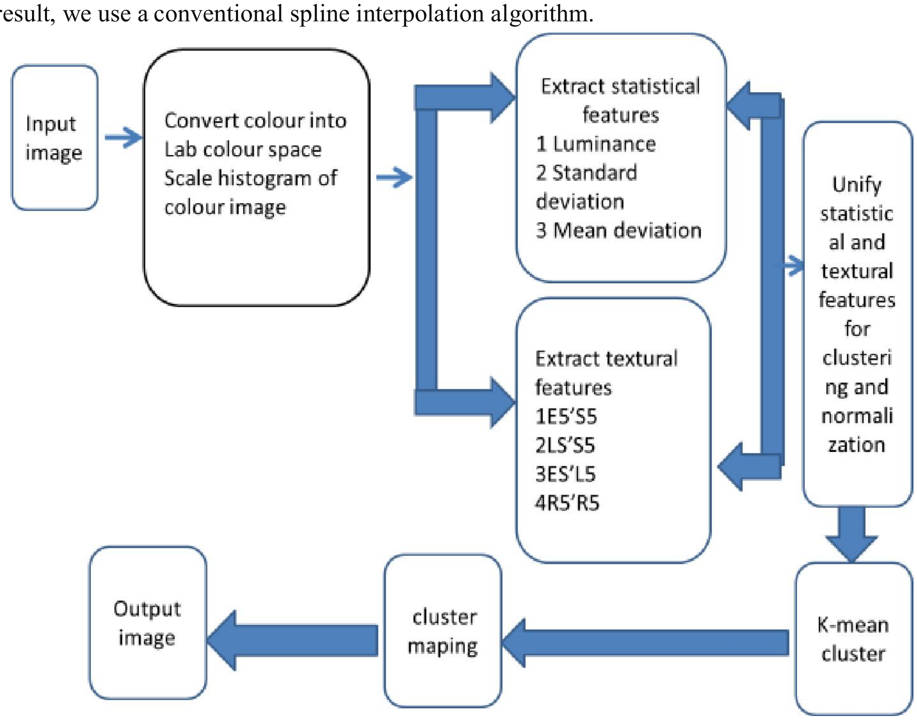

Fig 1: Process diagram

4.2 Image Processing

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Image processing is any form of processing for which the input is an image or a series of images or videos, such as photographs or frames of video. The output of image processing can be either an image or a set of characteristics or parameters related to the image. It also means Analyzing and manipulating images with a computer. There are some examples which are processed on image processing.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Impact Factor: 6.252
Volume 2, Issue 2, December 2022

Examples:

1. Normalization
2. Edge Filters
3. Soft focus, selective focus
4. Binning
5. User-specific Filter

A. Image Segmentation

Segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region is similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s) when applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of algorithms like Marching cubes.

B. Local Directional Pattern (LDP)

LDP is a gray-scale texture pattern which characterizes the spatial structure of a local image texture. A LDP operator computes the edge response values in all eight directions at each pixel position and generates a code from the relative strength magnitude. Since the edge responses are more illumination and noise insensitive than intensity values, the resultant LDP feature describes the local primitives including different types of curves, corners, and junctions, more stably and retains more information. Given a central pixel in the image, the eight directional edge response values { }, 0,1,...,7 i m i= are computed by Kirsch masks i M in eight different orientations centered on its position. The masks are shown in below figure. The response values are not equally important in all directions. The presence of a corner or edge causes high response values in some directions. Therefore, we are interested in the k most prominent directions to generate the LDP

-3	-3	5
-3	0	5
-3	-3	5

5	5	5
-3	0	-3
-3	-3	-3

5	-3	-3
5	0	-3
5	-3	-3

-3	-3	-3
-3	0	-3
5	5	5

Fig 2: Kirsch Edge masks in all four Directions

Here, the top k directional bit responses i b are set to 1.The remaining (8-k) bits of the 8-bit LDP pattern are set to 0. Finally, the LDP code is derived using equation 1. Figure3.6 shows the mask response and LDP bit positions, and Figure3.6 shows an exemplary LDP code with k=3. Local Directional Pattern (LDP) is a descriptor used for face recognition. It assigns a code for each pixel in the image, and the resultant LDP- encoded image is divided into regions for which each a histogram is generated. The significance of DR-LDP is the compact code generation for efficient face recognition.

85	32	26
53	50	10
60	38	45

→

313	97	503
537	x	393
161	97	161

Fig. 3 LDP Code with K=3

LDP = bi(mi– mk)x2b(a)=1

a>=0

0

a<0

Where, m k is the k-th most significant directional response. Since edge responses are more stable than intensity values, LDP pattern provides the same pattern value even presence of noise and non-monotonic illumination changes.
After computing the LDP code for each pixel (r , c) ,the input image I of size M × Nis represented by a LDP histogram H using equation 3. The resultant histogram H is the LDP descriptor of that image.
The LBP operator labels the pixels of an image by thresholding a 3x3 neighborhood of each pixel with the center value and considering the results as a binary number, of which the corresponding decimal number is used for labeling. The derived binary numbers are called local binary patterns or LBP codes. While the LBP operator uses the information of intensity changes around pixels, LDP operator use the edge response values of neighborhood pixels and encode the image texture. The LDP is computed as follow [24, 25]. The LDP assigns an 8 bit binary code to each pixel of an input image. This pattern is then calculated by comparing the relative edge response values of a pixel by using Kirsch edge detector. Given a central pixel in the image, the eight- directional edge response values are computed .Since the presence of a corner or an edge shows high response values in some particular directions.

4.3 Histogram

Histogram Equalization is a computer image processing technique used to improve contrast in images. It accomplishes this by effectively spreading out the most frequent intensity values, i.e. stretching out the intensity range of the image. An image histogram is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution at a glance. Given that their running times are generally in orders of minutes that makes them not suitable for colorizing too many images

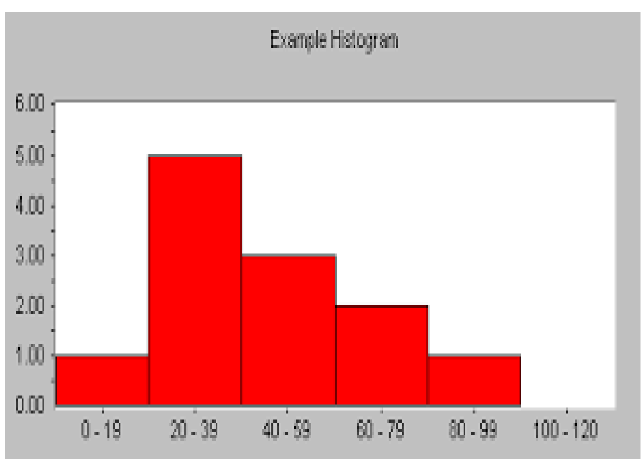

Fig. 4: Histogram Representation

4.4 Local Binary Pattern Histogram (LBPH)

LBPH (Local Binary Pattern Histogram) is a Face-Recognition algorithm it is used to recognize the face of a person. It is known for its performance and how it is able to recognize the face of a person from both front face and side face.

Local Binary Pattern (LBP) is an effective texture descriptor for images which thresholds the neighboring pixels based on the value of the current pixel. LBP descriptors efficiently capture the local spatial patterns and the gray scale contrast in an image. For calculating the LBP, the LBP code for each pixel is calculated and the histogram of LBP codes is constructed as the LBP feature.

To calculate the lbp code, for each pixel p, the 8 neighbors of the center pixel are compared with the pixel p and the neighbors x are assigned a value 1 if $x \geq p$. LBP feature vector, returned as a 1-by-N vector of length N representing the number of features. LBP features encode local texture information, which you can use for tasks such as classification, detection, and recognition. The function partitions the input image into non-overlapping cells. In addition to face and facial expression recognition, the LBP has also been used in many other applications of biometrics, including eye localization, iris recognition, fingerprint recognition, palm print recognition, gait recognition and facial age classification. So to find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.

So the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a 'confidence' measurement. Note: don't be fooled about the 'confidence' name, as lower confidences are better because it means the distance between the two histograms is closer.

Eigenfaces and Fisherfaces take a somewhat holistic approach to face recognition. You treat your data as a vector somewhere in a high-dimensional image space. We all know high- dimensionality is bad, so a lower-dimensional subspace is identified, where (probably) useful information is preserved. The Eigenfaces approach maximizes the total scatter, which can lead to problems if the variance is generated by an external source, because components witha maximum variance over all classes aren't necessarily useful for classification. How many images do we actually need to get such useful estimates? Here are the Rank-1 recognition rates of the Eigenfaces and Fisherfaces method on the AT&T Face database, which is a fairly easy image database.

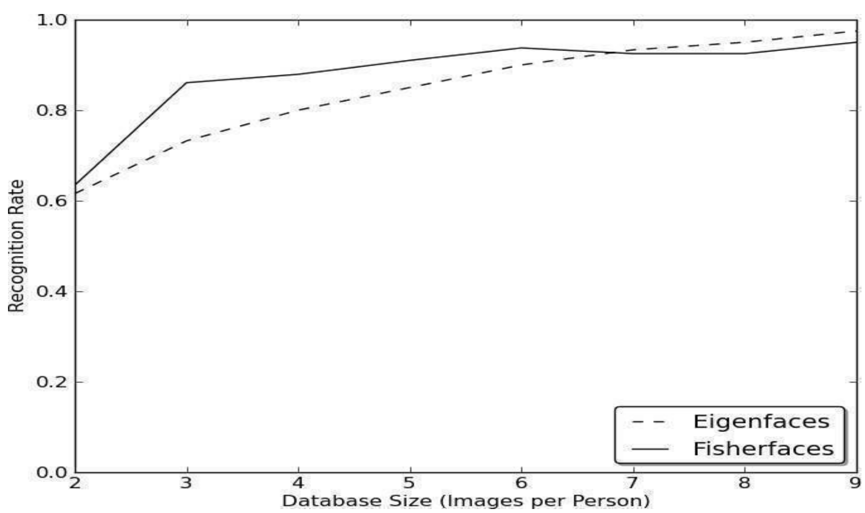

Fig.5: LBPH pattern in Graphical form

4.5 Grey Scaling

Gray scaling is the process of converting an image from other color spaces e.g. RGB, CMYK, HSV, etc. to shades of gray. It varies between complete black and complete white a series of regularly spaced tones ranging from black to white through intermediate shades of gray also an image composed solely of gray scale tones. The main reason why grayscale representations are often used for extracting descriptors instead of operating on color images directly is that grayscale simplifies the algorithm and reduces computational requirements. Grayscale images are distinct from one-bit bi-tonal black-and-white images, which, in the context of computer imaging, are images with only two colors: black and white (also called bi-level or binary images). Grayscale images have many shades of gray in between. Grayscale images can be the result of measuring the intensity of light at each pixel according to a particular weighted combination of frequencies (or wavelengths), and in such cases they are monochromatic proper when only a single frequency (in practice, a narrow band of frequencies) is captured. The frequencies can in principle be from anywhere in the electromagnetic spectrum.

4.6 Image Enhancement

Image enhancement is the procedure of improving the quality and information content of original data before processing. Common practices include contrast enhancement, spatial filtering, density slicing, and FCC. Image enhancement is the process of adjusting digital images so that the results are more suitable for display or further image analysis. For example, you can remove noise, sharpen, or brighten an image, making it easier to identify key features. The aim of image enhancement is ~~to~~ improve ~~the~~ interpretability or perception of information in images for human viewers, or to provide `better' input for other automated image processing techniques. Principle objective of Image enhancement is to process an image so that result is more suitable than original image for specific application. Digital image enhancement techniques provide a multitude of choices for improving the visual quality of images. Given that their running times are generally in orders of minutes that makes them not suitable for colorizing too many images.

However, with the K-mean clustering boom described in Section some researchers started tackling this to a higher degree of success as a fully automated process by training K- mean clustering's on large datasets such as SUN or Image Net. Currently, these methods are the state-of-the-art for natural image colorization. It is worth noting that even the application of CNNs to this task can be viewed as a form of automatic color or style transfer - with the references automatically pre-selected by the choice of the original CNN training set - using a complex method learned and realized.

The histogram graphically shows the following: skewness of the data; presence of outliers. A histogram is a chart that plots the distribution of a numeric variable's values as a series of bars. Each bar typically covers a range of numeric values called a bin or class; a bar's height indicates the frequency of data points with a value within the corresponding bin. It helps to visualize the distribution of the data.

Demerits are:

1) Cannot read exact values because data is grouped into categories.
2) More difficult to compare two data sets.
3) Use only with continuous data. A histogram is an approximate representation of the distribution of numeric data.

It varies between complete black and complete white a series of regularly spaced tones ranging from black to white through intermediate shades of gray also an image composed solely of gray scale tones. The main reason why grayscale representations are often used for extracting descriptors instead of operating on color images directly is that grayscale simplifies the algorithm and reduces computational requirements.

We all know high- dimensionality is bad, so a lower-dimensional subspace is identified, where (probably) useful information is preserved. The Eigenfaces approach maximizes the total scatter, which can lead to problems if the variance is generated by an external source, because components with a maximum variance over all classes aren't necessarily useful for classification. So to preserve some discriminative information we applied a Linear discriminate Analysis and optimized as described in the Fisherfaces method. The Fisherfaces method worked great at least for the constrained scenario we've assumed in our model. The main reason why grayscale representations are often used for extracting descriptors instead of operating on color images directly is that grayscale simplifies the algorithm and reduces computational requirements. Grayscale images are distinct from one-bit bi-tonal black-and-white images, which, in the context of computer imaging, are images with only two colors: black and white (also called bi-level or binary images). Grayscale images have many shades of gray in between. Grayscale images can be the result of measuring the intensity of light at each pixel according to a particular weighted combination.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
IJARSCT
Impact Factor: 6.252
Volume 2, Issue 2, December 2022

V. RESULTS AND DISCUSSIONS

Opening Interface:

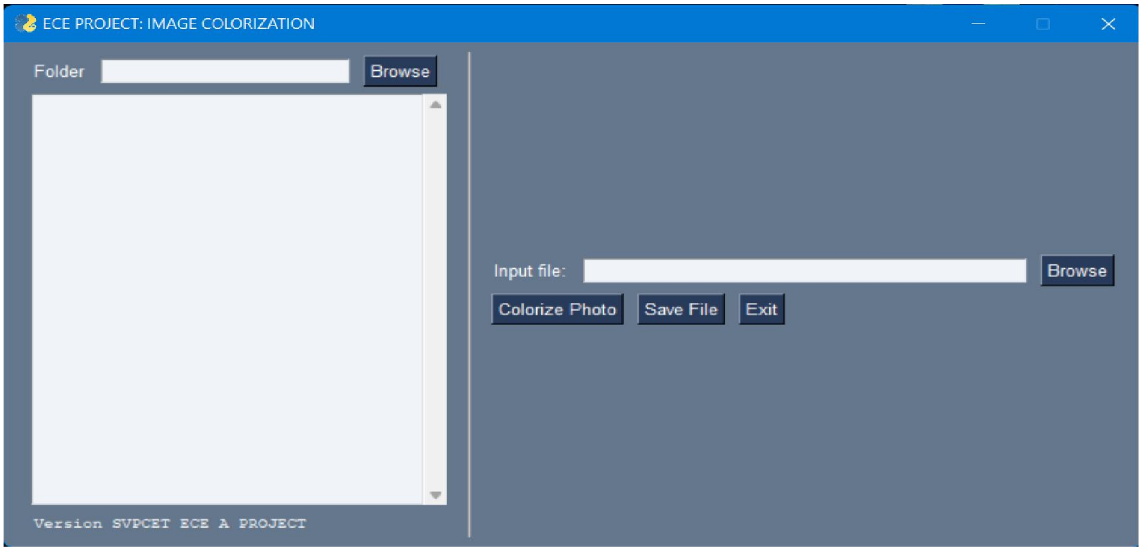

Fig.6: Initial interface

Figure representing the first interface of the application. It is the first step in the output window.

Selecting an image file:

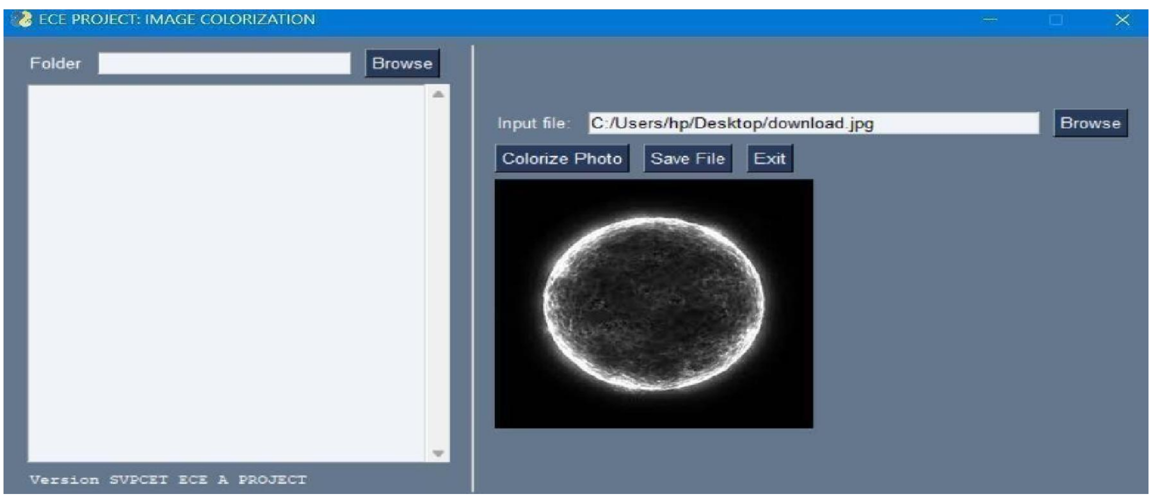

Fig. 7: Selecting an image file

By clicking on the "Browse" option, an image file can be selected from the storage. The selected file can be viewed in the application layout. It is the second step for colourizing the input greyscale image.

Converting image:

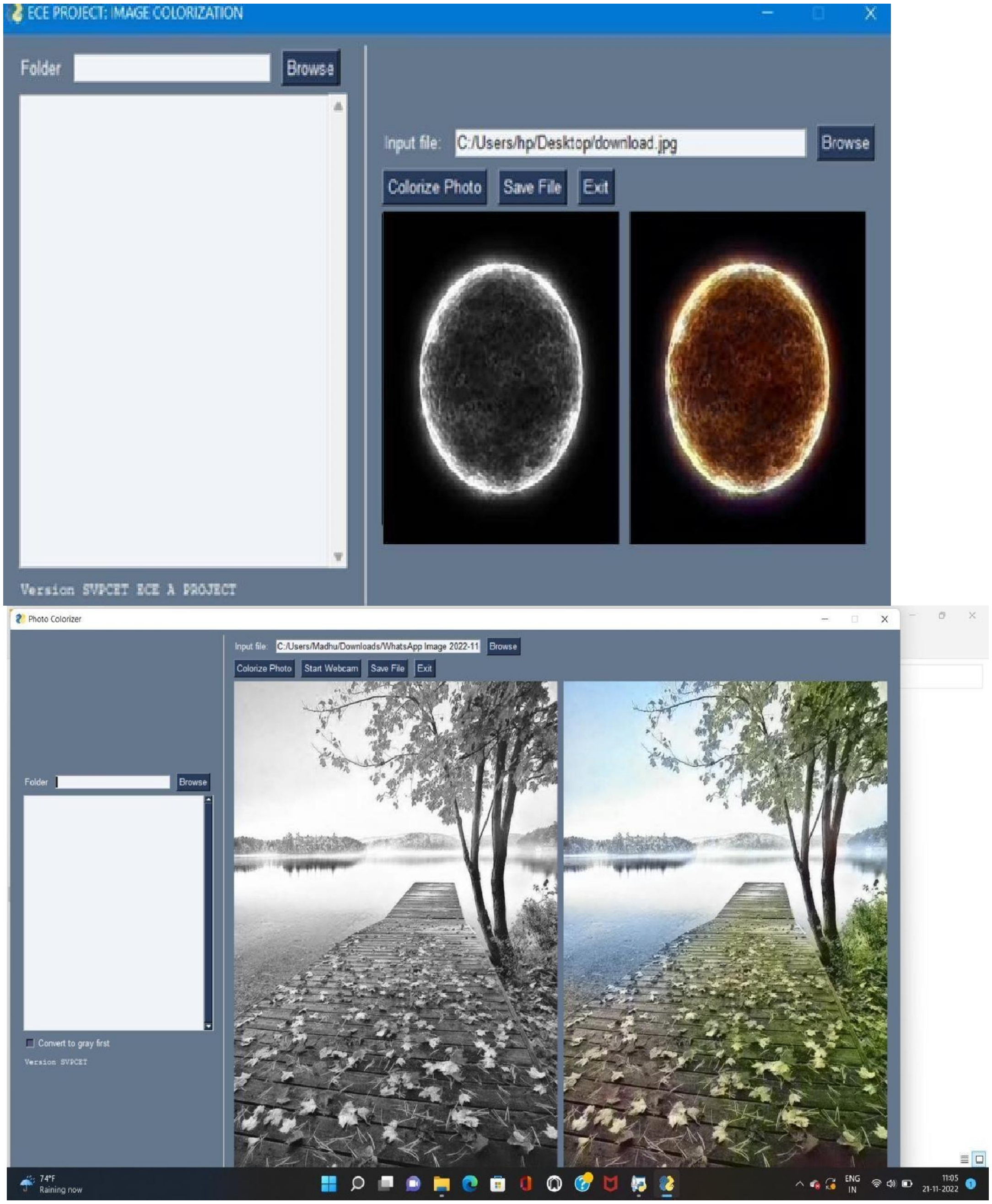

Fig.8: Image Outcome

On clicking on the "Colorize image" button, the application starts converting the grayscale image. The respective outcome of the picture will be displayed next to the input image.

VI. CONCLUSION

We presented a method of fully automatic colorization of unique grayscale images combining K-mean clustering techniques. Using color representation, we have shown that the method is capable of producing a plausible and vibrant colorization of certain parts of individual images even when applied to a moderately sized dataset that has properties which make it harder to colorize than natural images, but does not perform as well when applied to video sequences. In doing so, we visually and quantifiably compared several variants of K-mean clustering design, which differed in loss functions, architectures and regularization methods. It is clear that the models we used have a hard time learning colorization of large uniform regions such as background sky or walls but fare better when smaller objects and characters are present. We also proposed two methods of improving the generated results which greatly increase the visual resemblance of generated colorization to the ground truth images. One novel contribution is using and comparing a model inspired by residual K-mean clustering's for the task of colorization and showing that despite the smaller ERF and fewer parameters, it can generate results that are comparable or even surpass plain K-mean clustering in generalization to unseen data.

VII. FUTURE SCOPE

In order to be applicable for video, the method would currently require further refinement, performed manually by an artist. If trained on a larger dataset, the predictive power of the model would increase and is likely to produce more consistent colorization. For future work, it would be interesting to compare colorization produced by new models with significantly more depth (which require more computational resources to train) and models based on conditional generative adversarial networks, as the results they have been able to put together when applied to natural images are quite impressive and allow the userto have more control over the result by adjusting the latent space variable. Additionally, the K-mean clustering model could be adjusted to generate scribbles to use in conjunction with the algorithms, instead of full colorization. This could lead to results that more closely match the currently used colorization methods that apply color to greyscale images.

REFERENCES

[1]. Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. International Journal of Computer Vision, 2020

[2]. Image Colorization: A Survey and Dataset, Saeed Anwar, Muhammad Tahir, Chongyi Li, Ajmal Mian, Fahad Shahbaz Khan, Abdul Wahab Muzaffar,2020

[3]. Geirhos et al. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. ICLR, 2019.

[4]. Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 0–0, 2019.

[5]. Fang Liu, Xiaoming Deng, Yu-Kun Lai, Yong-Jin Liu, Cuixia Ma, and Hongan Wang. Sketchgan: Joint sketch completion and recognition with generative adversarial network. In CVPR, pages 5830–5839, 2019.

[6]. Hyunsu Kim, Ho Young Jhoo, Eunhyeok Park, and Sungjoo Yoo. Tag2pix: Line art colorization using text tag with secat and changing loss. In ICCV, pages 9056–9065, 2019.

[7]. Tsai-Ho Sun, Chien-Hsun Lai, Sai-Keung Wong, and YuShuen Wang. Adversarial colorization of icons based on contour and color conditions. In MM, pages 683–691, 2019.

[8]. Seungjoo Yoo, Hyojin Bahng, Sunghyo Chung, Junsoo Lee, Jaehyuk Chang, and Jaegul Choo. Coloring with limited data: Few-shot colorization via memory augmented networks. In CVPR, pages 11283–11292, 2019.

[9]. Bo Zhang, Mingming He, Jing Liao, Pedro V Sander, Lu Yuan, Amine Bermak, and Dong Chen. Deep exemplar based video colorization. In CVPR, pages 8052–8061, 2019.

[10]. https://www.python.org > to download the latest version of python.

[11]. https://numpy.org

[12]. https://pypi.org

[13]. Gokula Chandar ,Leeban Moses M; T. Perarasi M; Rajkumar; "Joint Energy and QoS-Aware Cross-layer Uplink resource allocation for M2M data aggregation over LTE-A Networks", IEEE explore, doi:10.1109/ICAIS53314.2022.9742763

[14]. Mustafa Alper Akkaş, Radosveta Sokullu, "An IoT-based greenhouse monitoring system with Micaz motes", https://doi.org/10.1016/j.procs.2017.08.300.

[15]. Dhuddu Haripriya, Venkatakiran S, Gokulachandar A, "UWB-Mimo antenna of high isolation two elements with wlan single band-notched behavior using roger material", Vol 62, Part 4, 2022, Pg 1717-1721, https://doi.org/10.1016/j.matpr.2021.12.203.

[16]. Gokula Chandar A, Vijayabhasker R., and Palaniswami S, "MAMRN – MIMO antenna magnetic field", Journal of Electrical Engineering, vol.19, 2019.

BIBLIOGRPHY

R.RAGHAVI
UG Student,
Dept Of Ece,Svpcet
Area Of Interest-
WirelessSensor

J. PRIYANKA,
UG Student,
Dept Of Ece,Svpcet
Area Of Interest-
WirelessSensor

K.THARUN,
UG Student,
Dept Of Ece,Svpcet
Area Of Interest-
WirelessSensor

T. PERUMAL REDDY,
UG Student,
Dept Of Ece,Svpcet
Area Of Interest-
WirelessSensor

C.NAVEEN,
UG Student,
Dept Of Ece,Svpcet
Area Of Interest-
WirelessSensor

K.SIVA PRIYA,
UG Student,
Dept Of Ece,Svpcet
Area Of Interest-
WirelessSensor