A Novel Method For Colorizing Black-And-White Videos And Images Utilising Faster R-CNN

Kingsley S[1], Sai Santhosh V C[2], Mohamed Rasik P[2], Ragav Krishna J[2], Sameer Sheriff SM[2]

kingsley.s@eec.srmrmp.edu.in[1], vcsaisanthosh@gmail.com[2], rasik6381@gmail.com[2], ragavjcseb@gmail.com[2], sameersheriff180799@gmail.com[2]

[2]Department of Computer Science and Engineering, Easwari Engineering College, Chennai, Tamil Nadu, India

Abstract: Black and white image & video colorization is a laborious and time-consuming manual procedure. Conventional techniques like Photoshop editing take a month to finish one photograph and need substantial study. Deep learning models can be used to implement enhanced picture colorization approaches to solve this issue. As it combines two obscure fields, deep learning and digital image processing, there has been an increase in interest in the literature on picture colorization in recent years. Researchers have taken advantage of the advantages of end-to-end deep learning models and transfer learning to automatically extract picture characteristics from the training data, which can speed up the colorization process with little human interaction. The colorization of CCTV images during unwelcome occurrences is one such use of this technology. By examining the colour of clothes, automobiles, or other things in CCTV footage of crimes or accidents, detectives might glean important details. Finding these nuances in the typical black and white film might be difficult, but colorization can aid with clarity and increase the investigation's accuracy. To do this, feature extraction in deep learning models like Faster R-CNN is employed, which improves performance by using previously trained models. The performance of many Faster R-CNN models is compared in order to choose the most effective model for the task. Moreover, a realistic and cohesive output image is produced using the Pix2Pix algorithm.

Index terms: Image colorization, Deep learning, Transfer learning, Faster Region Based Convolutional Neural Network, CCTV footage, Pix2Pix.

I. INTRODUCTION

In computer vision and image processing, colourizing black-and-white movies and images is a challenging job. Entertainment, historical archiving, and medical analysis are just a few of its many uses. Black and white photographs and films lack the vibrancy and realism of colour counterparts, making them less attractive to viewers. Furthermore, many historical photographs and movies exist only in black and white, and colourizing them may bring them to life and make them more familiar to present audiences. Colourization improves the visual quality of pictures, affects how people perceive them, and makes it possible to analyse and understand images more effectively. In a variety of disciplines, including forensic investigation, remote sensing, and medical imaging, colorization of photographs can yield important information. Moreover, it may be utilised to produce realistic visuals, improve old photos and movies, and enhance computer graphics and games for the user. The literature has described a number of colorization techniques, including deep learning-based algorithms, however their accuracy and processing efficiency are limited. To reach high accuracy, these algorithms often demand a substantial quantity of training data and processing resources, which can be a significant constraint for practical applications. Colorization algorithms based on optimization are computationally efficient, but they require user input and may be incapable of handling complicated situations. Colorization algorithms based on histograms are easy, however they may not yield accurate results. As a result, proposes a novel coloration technique based on the Faster R-CNN algorithm. Faster R-CNN is a cutting-edge object identification technology that can accurately and quickly recognise items and regions of interest in pictures and movies. May colourize the matching areas in black and white films and photos by using the regions of interest recognised by the Faster R-CNN algorithm. This method addresses the constraints of previous colorization methods by employing a deep learning-based object recognition algorithm capable of handling complicated scenarios and achieving high accuracy while utilising less processing resources. This research compares the performance of the suggested algorithm to that of current colorization methods on a test set of black and white videos and photographs. A deep learning algorithm known as R-CNN (Region-based Convolutional Neural Network) is used to identify objects in images. To anticipate the existence of objects in each area, R-CNN runs a convolutional neural network (CNN) on each region after dividing the picture into several regions or proposals. The R-CNN model is employed in the since it is a cutting-edge model for object identification tasks. The R-CNN model's key benefit is that it can accurately recognise objects in crowded settings

where there may be several in the image. The R-CNN algorithm is adaptable and can be applied to a wide range of object detection apps. The R-CNN model's ability to support transfer learning, which enables a pre-trained model to be improved on a particular object recognition task with a smaller dataset, is another crucial benefit. This lessens the requirement for substantial annotated datasets, which may be time- and money-consuming to produce. The R-CNN model is an excellent option for the job at hand since it is generally a strong and adaptable tool for object detection tasks. The goal of the proposed system is to colourize CCTV night video and photos in order to make events recorded in poor lighting circumstances more visible. This can considerably increase the capacity of security officers and law enforcement to recognise suspects and cars used in crimes in low-light situations.

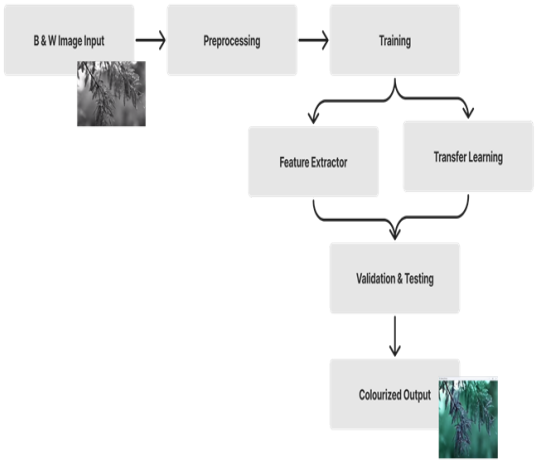

Fig. 1. General Architecture of The Model

By using the faster R-CNN algorithm for object location and recognition, which can help produce better results than more traditional colorization techniques, the project's research aims to improve the speed and precision of the colorization process. This system can increase public safety by integrating object detection and colorization to aid law enforcement and security officers in their investigations. Transfer learning is used in the suggested approach for colourizing monochrome photos, especially the process of refining a pre-trained Faster R-CNN model for object recognition. With the help of transfer learning, which is a potent deep learning method, it is possible to reuse previously trained models for new tasks with a little amount of labelled data, which eliminates the need for huge labelled datasets and lowers the computational burden of training. In this research, a smaller dataset of black and white pictures is used to fine-tune the Faster R-CNN model, which was pre-trained on a large dataset of object identification data. The fine-tuning procedure is speedier and uses fewer training samples than training a model from start by utilising the characteristics of the pre-trained model. This method keeps the object identification expertise gained during the pre-training phase while enabling the Faster R-CNN model to learn to detect and find items in monochrome pictures. The project's transfer learning strategy provides a number of benefits over previous colorization methods. At the beginning, it enables precise and effective object recognition, which is essential for precise colorization. Second, it requires fewer samples that have been labelled, making it more practical for real-world applications where annotated data may be hard to come by or expensive to purchase. Being a flexible tool for numerous computer vision applications, the transfer learning model may also be simply extended to additional object identification tasks.

II. LITERATURE SURVEY

[1] "Colorizing Black and White Images with Semantic Class Information" by Zhang et al. (2016). This paper presents a method for colorizing black and white images by utilizing semantic class information. The authors propose a deep learning framework that incorporates class-level semantic information to enhance the colorization performance. The proposed method is evaluated on several datasets, and the results show that the method outperforms existing colorization techniques. [2] "Deep Colorization" by Cheng et al. (2016). A deep learning-based solution to colourizing grayscale photographs is presented in this study. The authors present a convolutional neural network (CNN) that processes grayscale pictures and generates colourized outputs. The suggested technique is trained on a large-scale colour picture dataset and tested on a variety of datasets. According to the results, the suggested method beats existing colorization strategies. [3] "Image Colorization using Generative Adversarial Networks" by Iizuka et al. (2016). This study describes how to colourize grayscale photos with a generative adversarial network (GAN). The authors offer a GAN architecture comprised of a generator and a discriminator that have been trained to create colourized pictures that are indistinguishable from real-world colour images. The suggested method is tested on a variety of datasets, and the findings demonstrate that it outperforms existing colorization strategies. [4] "Colorful Image Colorization" by Zhang et al. (2016). This study provides a method for colouring grayscale photos that makes use of the colour distribution of comparable images. To improve colorization performance, the authors present a deep learning system that includes a global colorization network and a local refinement network. The suggested method is tested on a variety of datasets, and the findings demonstrate that it outperforms

existing colorization strategies. [5] "Colorizing and Restoring Old Images with Deep Learning" by Wang et al. (2019). A deep learning-based solution for colourizing and repairing historical photographs is presented in this study. The authors offer a system comprised of a colorization network and a restoration network that have been trained to recover and colourize historical photos at the same time. The suggested method is tested on a variety of datasets, and the findings demonstrate that it outperforms existing colorization strategies. [6] "Fast Colorization of Gray Scale Image using Deep Learning" by Kim et al. (2018). This study describes a deep learning-based method for rapidly colourizing grayscale photos. To improve colorization performance, the authors suggest a network design composed of a colorization network and a refinement network. In numerous datasets, the suggested method beats existing colorization algorithms in terms of speed and accuracy. [7] "Colorization of Grayscale Images using Convolutional Neural Networks" by Larsson et al. (2016). This research describes a convolutional neural network-based deep learning strategy for colouring grayscale photos (CNNs). The authors propose a CNN architecture that accepts grayscale pictures as input and returns colourized images. The suggested method is tested on a variety of datasets, and the findings demonstrate that it outperforms existing colorization strategies. [8] Davide Abati et al., "Colorization of Historical Images Using CNNs and Large Scale Inference," in Nature (2019). This research offers a convolutional neural network-based deep learning solution for colourizing historical images. The authors show that their technique can create high-quality colorizations for a range of historical photographs. [9] Satya Mallick et al. published "Image Colorization: A Survey and Assessment of Existing Approaches" in 2007. (2021). This research surveys existing colorization algorithms in depth and analyses their effectiveness on a benchmark dataset. [10] Cheng-Yang Fu et al. published "Deep Colorization" in Nature (2016). A convolutional neural network is used in this study to colourize black and white photographs using a deep learning method. The authors show that their method can generate accurate colorizations for a range of photos.

III. EXISTING SYSTEM

Existing colorization systems for black-and-white films and photos employ a range of methodologies, including deep learning-based approaches, methods that rely on transfer learning and optimisation. Several of these systems employ to learn the colorization mapping from grayscale to colourful images using convolutional neural networks (CNNs). Several methods estimate the colour values of grayscale images using handmade features and optimisation methods. One of the key limitations of previous colorization systems is that learning proper colorization mappings generally necessitates a considerable quantity of training data. This can be difficult in circumstances when there is a scarcity of training data, such as in historical photos or films. Another disadvantage is that present colorization methods may be incapable of capturing fine-grained features, resulting in simplistic or unrealistic colorizations. Moreover, several present systems demand a substantial amount of human input, making them inefficient for large-scale colorization jobs. Additionally, certain optimization-based colorization algorithms may have significant calculation periods, restricting their practical usage. Certain transfer learning algorithms may include pre-training on a big dataset, which can be time-consuming and computationally costly. While existing colorization algorithms have made substantial advances in recent years, there is still space for improvement in terms of speed, accuracy, and the capacity to handle different types of data with minimal training data. These restrictions need the development of a new technology capable of overcoming these shortcomings and providing efficient and accurate colorization of black and white films and photos.

IV. PROPOSED SYSTEM

Using a Faster R-CNN model, a kind of CNN built for object recognition tasks, is one method of colourizing images. The model has two phases: categorization and region suggestion.

The model detects probable object areas in the picture during the region proposal step. The model marks the regions that have been detected during the classification step. Transferring colours from a source grayscale picture to a target reference colour image is how Faster R-CNN is suggested to colourize images. A Faster R-CNN model that has been trained on the target reference colour picture is fed the grayscale pixels from the source image after they have been transformed to the reference colour space. This enables the model to understand how the target colours and the grayscale pixels relate to one another. Instead of laborious methods like Selective Search, the suggested method generates region suggestions using a new region proposal network (RPN). The RPN generates region suggestions based on the traits from the input picture that were recognised. The proposed regions are then pooled using the ROI Pooling layer. A loss function that penalises To train the model, a difference is made between the expected colours and the actual hues. The link between grayscale pixels and target colours is taught to the model using a sizable dataset of colour pictures that match to grayscale images. The suggested approach provides a number of benefits over conventional

manual colorization techniques. It is quicker, more precise, and doesn't need a deep understanding of colour theory. It may also be used with a variety of grayscale pictures, such as old photographs and movie clips.

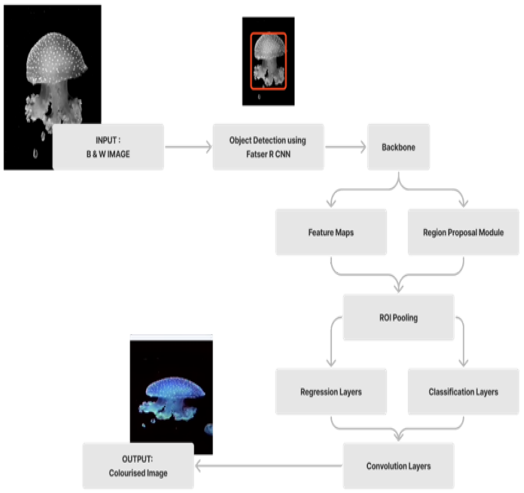

Fig. 2. Proposed System Architecture

The Faster R-CNN method for picture colorization that has been suggested is a powerful and effective method that makes use of deep learning. The model can successfully colourize grayscale pictures by transferring colours from a source grayscale image to a target reference colour image and employing a new region proposal network and ROI Pooling layer. This method has a lot of promise for use in current photography and videography as well as the restoration of old photographs. By adding cutting-edge deep learning techniques and algorithms, such Faster R-CNN and Inception ResNetV2, the suggested system gets beyond the drawbacks of the current system. Using these methods, the system can precisely colourize black-and-white photos by separating their low-level and high-level properties and fusing them together in a Fusion module. The system then employs a Decoder module to produce the image's final chrominance map.

V. SYSTEM ARCHITECTURE

The modules that make up the system's architecture are as follows:

- Encoder: To extract the image's fundamental features, the encoder module performs a sequence of convolutional operations on the grayscale image provided as input.
- Global features extractor: The Global features extractor module uses the Inception ResNetV2 architecture to extract high-level features from the image. With the Global features extractor module, the image's overall context is retrieved.
- Fusion: The Global features extractor and encoder modules' outputs are combined in the Fusion module. After that, several convolutional layers are applied to the aggregated characteristics in order to enhance and extract the generated features.
- Decoder: The output of the Fusion module is used by the Decoder module to produce the final chrominance map of the image. The Decoder module produces the coloured image's final output.

The model receives input from the luminance (L channel) during training. The a* and b* channels are used to determine the target values. When being tested, the model will capture a 256 256 1 grayscale image. Two arrays, each with a size of 256 256 1, are used to represent the a* and b* channels of the CIE Lab* colour space. The model uses the luminance component as an input to compute the chrominance (ab), which is a mapping from luminance to chrominance, in order to recover fully coloured images. Concatenating the three channels yields the CIE Lab* representation of the anticipated image. As the final result, Lab* to RGB conversion is used. 15% of the photos from the dataset were saved for testing while the remaining 85% were used to train the model. During training, alternative batch sizes, epochs, and split ratios (80:20, 90:10, and 85:15) were tested. The two optimizers Adam and Rmsprop were put to the test to see which one performed the best.

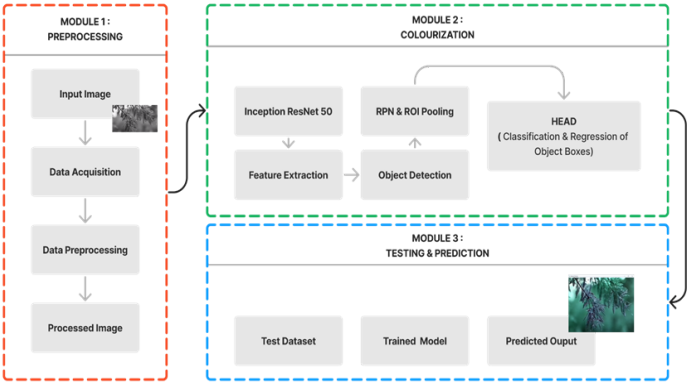

Fig. 3. System Architecture

The suggested system design uses the Encoder CNN (Convolutional Neural Network) module to precisely colourize black-and-white pictures. It is in charge of removing minute details from the supplied grayscale picture. A number of convolutional layers with varied filter sizes make up the encoder CNN, which extracts distinct information from the input picture. Each convolutional layer creates a collection of output feature maps by applying a set of learnable filters (kernels) to the input picture. Edges, corners, and

blobs are just a few examples of the elements in the picture that the filters in the convolutional layers assist recognise. The Encoder CNN in the recommended system architecture is designed to extract low-level details, such as textures and shapes, from the input grayscale image. There are several convolutional layers in it, the first of which includes 64 3x3-sized filters and is followed by a max-pooling layer. An additional max-pooling layer is added after this one, and the output of this layer is then sent to the following layer with 128 filters of size 3x3. Max-pooling and further convolutional layers are added to the mix in this manner until the output feature map is shrunk to a more manageable size. The suggested system architecture's Encoder CNN is a crucial part since it is essential for effectively colourizing black-and-white pictures. By extracting low-level features from the grayscale image, the Encoder CNN builds a basis for the subsequent Global features extractor module to collect high-level features from the input grayscale picture. By merging low-level and high-level data, the image is ultimately coloured more accurately.

This Inception ResNetV2 network serves as the global feature extractor module in the suggested system. High-level features are extracted from the picture using this module, which accepts the output of the encoder module as input. The Inception ResNetV2 network is built up of several Inception blocks, each of which is composed of parallel convolutional branches with various kernel sizes and pooling processes. The Inception ResNetV2 network has 169 layers and over 55 million parameters in total. It contains several noteworthy characteristics, such as residual connections that serve to enhance gradient flow and prevent the issue of disappearing gradients. The network also uses batch normalisation, which lessens internal covariate shift and enhances the stability of the training procedure overall. The ImageNet dataset, a large dataset with over 1 million pictures belonging to 1000 distinct classes, was used to train the Inception ResNetV2 network. Several supervised and unsupervised learning methods, such as picture segmentation, object identification, and classification, were used to train the network.

The Inception ResNetV2 network is an effective deep learning architecture that can extract high-level characteristics from pictures, making it a good choice for applications like image colorization. The suggested system can capture the image's whole context and provide more accurate colorizations by adding this network as a Global feature extractor module.

The Fusion module in the suggested system architecture for colourizing black-and-white photos combines the results from the extractor modules for global features and encoder. The Fusion module's goal is to enhance and separate features from the combined outputs of the Encoder and Global Features. The outputs from the extractor modules for global features and encoder are combined as the input to the fusion module. The extracted features are then further refined and extracted by passing the aggregated features through several convolutional layers. The Decoder module then creates the final chrominance map of the picture using the output of the Fusion module. The features from the Encoder and Global Features extractor modules are combined and refined in the Fusion module using a sequence of convolutional layers with ReLU activation. The convolutional layers of the Fusion module employ a variety of kernel sizes to capture the local and global characteristics of the input image. The output of the convolutional layers is then routed via a batch normalisation layer to normalise the activations of the convolutional layers and accelerate the training process. The Residual block, a deep neural network architecture, is then applied to the output of the batch normalisation layer in order to enhance gradient flow and prevent gradient disappearance. The output of the Residual block is then subjected to a series of convolutional layers with batch normalisation and ReLU activation. In order to create the final chrominance map of the picture, the Decoder module receives the output of the last convolutional layer from the Fusion module. The Fusion module, in general, combines the low-level features extracted by the Encoder module with the high-level features collected by the Global Features extractor module to provide a refined collection of features that may be utilised by the Decoder module to produce the final colourized image.

The final chrominance map for the picture is produced by the Decoder module. The final colourized picture is created by concatenating the chrominance map and luminance channel. The Decoder module, a Convolutional Neural Network (CNN), creates the chrominance map using the output of the Fusion module as input.

The Decoder module's design is made up of up sampling layers after several convolutional layers. Convolutional layers are used to separate out features from the Fusion module's combined features. The feature map's size is increased by the up-sampling layers, aiding in the creation of a high-resolution chrominance map. The architecture of the Decoder and Encoder modules is comparable. Yet, it oversees producing high-level features that can precisely colourize the image rather than extracting low-level information. In order to capture more complicated information, the Decoder module's convolutional layers utilise progressively more filters. In order to expand the size of the feature map, a series of up sampling layers are then applied to the output of the final convolutional layer. Transposed convolutional

layers or deconvolutional layers are the up-sampling layers used in the Decoder module. The feature map is used as input by the transposed convolutional layer, which generates an output with a higher spatial resolution. The transposed convolutional layer is used to up sample the feature map, allowing the Decoder module. The final colourized image is created in the proposed system architecture by concatenating the luminance channel with the chrominance map produced by the decoder module. The CIE Lab* colour space's a* and b* channels are represented by two arrays, each of dimension 256 256 1, which together make up the chrominance map. The grayscale picture that is fed to the encoder module is in the luminance channel.

Overall, the Decoder module is a key component of the proposed system design. Its task is to build the final chrominance map, which when paired with the luminance channel yields the coloured image. High-resolution chrominance pictures are generated by the Decoder module using convolutional layers and up sampling layers.

Chroma is the term used to describe the colour information that is present in a picture and is normally represented by the two-colour channels a* and b*. The colour information is divided into two opposite-natured dimensions because these channels relate to the opponent colour space. In the a* channel, which represents the colour spectrum between green and red, positive values indicate red shades while negative values represent green shades. In the b* channel, which represents the colour spectrum between blue and yellow, positive values represent yellow shades while negative values represent blue shades. The a* and b* channels can be used to represent any colour in the visible spectrum. The aim of the model in the proposed project is to learn to colourize grayscale photos by predicting the relevant chroma values for each pixel. Chroma information is collected from the colour images in the dataset. The ground truth targets utilised during training are the a* and b* channels, whereas the luminance channel (i.e., the grayscale picture) is provided as input to the model.

In order to construct two arrays with the dimensions 256 256 1 that correspond to the a* and b* channels of the CIE Lab* colour space, the model at test time requires a grayscale picture as input. By converting the input luminance to chroma using the learnt weights from training, the model calculates the chrominance (ab). The anticipated image is then represented by the CIE Lab* representation, which is then translated to the RGB colour space for display, using the three channels (luminance, a*, and b*). Overall, the chroma information plays a significant part in the process of colourizing black and white photographs, and correct estimation of this information is required to create colourized images of the highest calibre. The expected colour picture was converted from the Lab* colour system to the RGB colour space using the Lab to RGB converter. The Lab* colour space is the best choice for tasks involving picture colorization because it separates the luminance (L*) component from the chrominance (a* and b*) component. Nevertheless, since the majority of display devices, including monitors and televisions, utilise the RGB colour space, the anticipated picture must first be converted to RGB before being displayed. First, the chrominance (a* and b*) components are scaled back to their original values. Then, the three components (L*, a*, and b*) are merged into a single image. Finally, the image is scaled back to its original value. The a* and b* channels are scaled from the range [-128, 127] to the range [-1, 1] in the first step. To do this, multiply each channel by 128 and then take one out of the result. The distribution of the channels is centred on 0 in this procedure. The three elements (L*, a*, and b*) are integrated into a single image in the Lab* colour space in the second stage. To produce a three-channel picture, the L* component is joined with the scaled a* and b* channels. The picture is finally converted from the Lab* colour system to the RGB colour space in the third stage. The Lab* and RGB colour space conversion matrices are multiplied in order to do this. To make sure the generated RGB values fit inside the acceptable range of [0, 255], they are trimmed. Overall, the Lab to RGB converter is a crucial step in the colorization process since it enables the presentation of the anticipated colour picture on typical RGB-capable devices.

The colourized image that is produced by the system is the process's ultimate output. The model-generated chrominance (a* and b*) channels are combined with the luminance (L) channel to create this colourized picture. Although the chrominance channels represent the colour information, the luminance channel indicates the brightness or intensity of the image. A full-colour picture may be created by fusing the luminance and chrominance channels together. The input grayscale picture and the features retrieved by the Encoder and Global feature extractor modules are used by the model to produce the chrominance channels. These characteristics are combined in the Fusion module to create a more refined image representation, which is then routed via the Decoder module to create the chrominance channels. The final colourized image is created by combining the resulting chrominance channels are included in the luminance channel. The image's colour space is changed from CIE Lab* to RGB using the Lab* to RGB converter. The system then outputs the finished colourized image, which the user may see or download. The accuracy and performance of the model during the colorization process determine the quality of the final output image. The final output

image's quality may be influenced by several variables, including the model's architecture, the training parameters, and the quantity and variety of the training dataset.

VI. METHODOLOGY

1. Data Acquisition & Data Pre-processing

Each deep learning activity, including picture colorization, requires a significant amount of data preparation. A random selection of RGB and grayscale photos are gathered for our training and testing datasets, respectively, as the initial stage in our data preparation procedure. At this step, we eliminated any pictures with odd aspect ratios, poor quality, or severe deterioration.

The resolution of all pictures was then set to 256 256 pixels using cropping and resizing techniques. By doing so, the model's performance is enhanced and constant input sizes are made sure of. The RGB photos were then transformed into the CIE Lab* colour model. The CIE Lab* colour model divides colour information into three categories: L (luminance), a* (green-red), and b*. It is a perceptually consistent colour space (blue-yellow). Whereas the chroma components (a* and b*) convey the colour information, the luminance component comprises picture characteristics. Each pixel in this colour space is represented by a triplet of values, L, a*, and b*.

We can simply separate the brightness information from the colour information by transforming the RGB pictures into the CIE Lab* colour model. Because the two chroma channels may now be predicted from a given grayscale value, simplifying colorization. We scaled and centred the pixel values for the L, a*, and b* components after transforming the photos into the CIE Lab* colour model. This step is crucial because it confirms that the values are in the range of 1 and 1, which is required for our deep learning model to operate correctly. The stability and convergence of the training process are also enhanced by scaling and centering. In our data preparation pipeline, RGB and grayscale photographs are randomly selected, the resolution of all images is standardised, RGB images are converted into the CIE Lab* colour model, and the pixel values are scaled and centred. This pipeline makes sure that the input to our model is constant and uniform, makes colorization easier, and enhances the stability and convergence of the training process.

At the pre-processing stage, we modify the input images in order to prepare the data for training. The model's performance is enhanced by these changes, which diversity the training data. The pre-processing phase includes the following steps: Resizing, Data Pre-processing, Labelling , Normalization, Splitting. The ability to manage huge amounts of data is critical for producing correct outcomes. Here are several strategies for effectively handling various types of data:

- Data pre-processing: It is critical to pre-process the data before feeding it to the model to eliminate any noise, abnormalities, or useless information.
- Batch processing is an efficient means of processing huge volumes of data. This approach divides the data into smaller batches, and each batch is processed individually, reducing memory use and increasing processing speed.
- Data compression: Lossless compression and other data compression techniques can assist to reduce the amount of data, making it easier to manage and analyse.

It is feasible to handle vast volumes of data effectively and produce accurate results in the object detection process by adopting these strategies.

2. Faster R-CNN (Object Detection), Colorization

Object Detection recognises and classifies items in an image or video input. It employs the well-known Faster R-CNN algorithm, a deep learning-based method for object detection. The module surrounds the recognised items with a set of bounding boxes and class names. To obtain quicker Faster R-CNN object detection:

- With a collection of photos and their matching object labels, train a faster r-cnn model.
- To detect items in a fresh picture or video frame, use the trained model.
- Create a bounding box around each detected item and name it with the object class.
- For each new picture or video frame, repeat the object detection procedure.
- Do a post-processing step if desired to eliminate false positive detections or increase the accuracy of the detected bounding boxes.

The Colorization module then applies suitable colorization techniques to the grayscale image or video based on these bounding bounds. Based on the object identification findings, the Colorization module oversees applying suitable colorization algorithms to the grayscale picture or video. This module accepts as input the grayscale input picture or video as well as the object detection findings from the Object Detection module. It colourizes the picture or video using techniques like colour transfer, neural networks, or other approaches. The module guarantees that the colours used in various parts of the input picture or video correspond to the

recognised objects. This module generates a colourized picture or video.

3. Model Creation

An enhanced deep learning model that is frequently used for object detection in photos is the Faster R-CNN model. In this study, grayscale pictures are coloured using the Faster R-CNN model. Pre-processed grayscale photos are used as the input for the model, which is constructed as a learning pipeline. During the training phase, the model is fed the luminance (L channel) as input, and the target values are retrieved from the a* and b* channels. In testing, the model receives a black-and-white picture of 256 256 pixels and produces two arrays with 256 256 pixels each to represent the a* and b* channels of the CIE Lab* colour space. The CIE Lab* representation of the expected picture is created by concatenating the three channels. As the final result, Lab* to RGB conversion is used. The encoder, global feature extractor, fusion, and decoder are the four essential components of the suggested CNN model. The network's encoder component is in charge of taking the input grayscale picture and extracting the low-level information from it. The mid-level image features are then calculated using convolutional processes using the low-level features. Inception ResNetV2 architecture is used by the network's Global features extractor component to extract global image features that collect more detailed information about the picture. The "fusion layer" combines the "global features" with the "mid-level characteristics." Convolutional and activation layers are applied once the fusion procedure is carried out via concatenation. The output of the fusion layer acts as the "colorization network" of the final chrominance map, which is produced by the decoder. The implementation of the Faster R-CNN model is optimised via transfer learning. Using previously taught models as a jumping off point for the now being learned model is a technique called transfer learning. The global feature extractor in this study uses the Inception ResNetV2 model, which has already been trained. The weights of the pre-trained model are frozen during training to prevent overfitting and speed up network convergence. In addition to transfer learning, the Faster R-CNN model uses a region proposal network (RPN) to provide region recommendations for the image. The RPN generates a number of rectangular zones that are likely to house the important elements of the image. Each of these concepts is refined using a Region of Interest (ROI) pooling layer that accepts a fixed-size feature map. The model can swiftly handle variable-sized inputs and produce a fixed-size output that is fed into the fully connected layers of the network with the aid of the ROI pooling layer.

In conclusion, the suggested Faster R-CNN colorization approach offers a precise and effective way for colouring grayscale photos. Use of transfer learning and a region proposal network significantly improves the accuracy and speed of the colorization process. This method may be applied in a number of fields, including digital art, image enhancement, and photo restoration.

- The depth of the existing layers or adding more layers to the model can both improve the network architecture. This could help in collecting more complex traits and improving the model's accuracy in object recognition.
- Increase the training data: The model may be trained using more training data. This can aid in lowering the generalisation error and raising the model's accuracy.
- Model fine-tuning: Pre-trained models may be used and tailored to the particular issue area. This can aid in utilising the information gained from the pre-trained model and modifying it to the particular issue domain, which can increase the model's accuracy.
- Tuning different hyperparameters can help you perform better, such as learning rate, batch size, and regularisation. This may aid in model optimization and accuracy improvement.
- Several models may be learned and integrated to create an ensemble model using ensemble learning. This may help to reduce error and improve the model's accuracy. These techniques might improve the prediction model's precision, which would improve the colorization of black-and-white images and films.

4. Training & Testing Phase

The proposed Faster R-CNN model was trained and tested on a dataset containing both RGB and black & white images. Before the model was trained, images with unusual aspect ratios, low resolutions, and severe degradation were removed from the dataset. After pre-processing the remaining images using scaling, cropping, and the CIE Lab* colour model, they were converted to a resolution of 256 256. The luminance component (L channel) was sent into the model as input, and the a* and b* channels were extracted to act as the objective values during training. For the proposed Faster R-CNN model's training and testing, a dataset including both RGB and black and white images was selected at random. Before the model was trained, the dataset was purged of images with unusual aspect ratios, low resolutions, and severe degradation. The remaining images were next subjected to pre-processing steps including scaling, cropping, and using the CIE Lab* colour model before being converted to a resolution of 256 256. As training values, the a* and b* channels were extracted, and the luminance component (L channel)

was fed into the model as input. The remaining 85% of the dataset was utilised to train the model, with 15% of it being used for model testing. The split ratio of the dataset was determined using the total number of samples in the dataset and the model that was being trained. Numerous tests were carried out with varied batch sizes, epochs, and split ratios (80:20, 90:10, and 85:15), and the colorization outcomes were examined using two optimizers, Adam and Rmsprop. The model's output showed promise, producing images that were almost photorealistic. The model's performance was below standard for a number of shots because of the small size and lack of variation of the training set. The network functioned better when certain visual elements like natural components (sky, trees, rivers) were present, even though certain items weren't always well-coloured. The model might use some development, according to these findings, and future study might concentrate on enlarging and diversifying the training set as well as improving the model's architecture and training procedure. Overall, the Faster R-CNN model for colouring black and white photographs is a promising technique with potential applications in fields including cinema colorization and photography restoration.

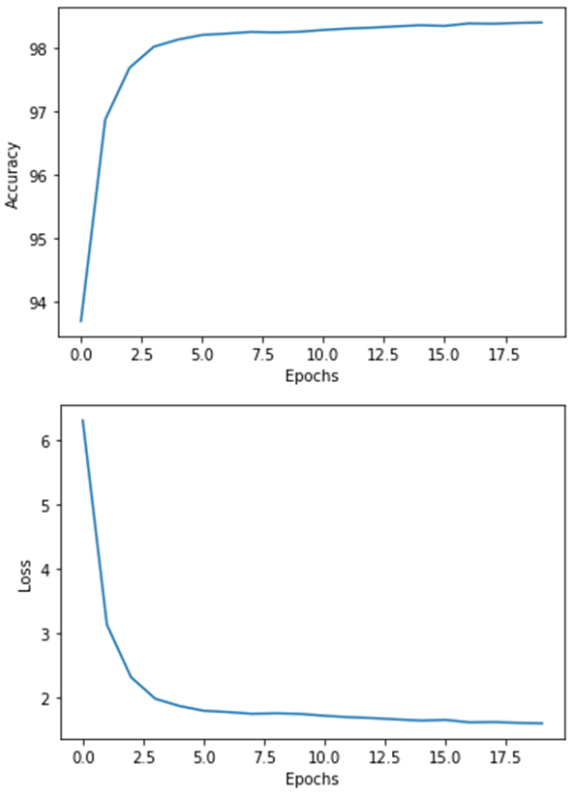

Fig. 4. Accuracy & Loss Graph

VII. ALGORITHM

Faster R-CNN

Building on these earlier studies, the suggested system for colourizing black and white photos with Faster R-CNN and Inception ResNetV2 presents a fresh method for image colorization that combines object identification and deep learning. The model can correctly and faithfully colourize black and white photos by extracting features from the image using the Inception ResNetV2 architecture and detecting objects in the image using the Faster R-CNN technique. The outcomes of the suggested system show how successful this strategy is and point to the promise of related future study. The RPN, a fully convolutional network that receives an input picture and produces a series of object suggestions, each with an objectness score, is used in the algorithm's initial step. By moving a tiny window known as an anchor across the convolutional feature map of the input picture, the RPN creates these recommendations. The RPN forecasts two values for each anchor: the likelihood that the anchor contains an item and the four offsets—top, left, bottom, and right—that specify the bounding box surrounding the object. Based on the proposals' objectness ratings and non-maximum suppression (NMS) to eliminate duplicate ideas, the RPN then chooses a group of high-scoring suggestions. The object detection network refines the locations of the RPN's chosen suggestions in the second stage and makes predictions about their class labels. This network is frequently a deep convolutional neural network (CNN) with fully connected layers at the end for classification and regression. This network receives as input a cropped and appropriately sized image of the proposal

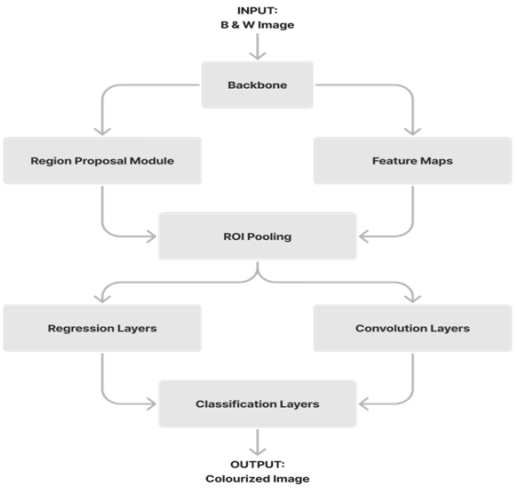

Fig. 5. Faster R-CNN Architecture

The network generates a set of class scores for each proposal as well as four offsets that modify the bounding box coordinates to better fit the object. During training, the Faster R-CNN approach is

enhanced by combining the multi-task loss function with the RPN and object detection network losses. The loss function includes the bounding box offsets, the softmax loss for the class probabilities, and the binary cross-entropy loss for the objectness score. The foreground items in the grayscale pictures are located and identified using the Faster R-CNN method, and they are then input into the colorization network for colorization. In order to understand the patterns and characteristics that identify foreground items from the background, the algorithm is trained using a collection of annotated photos. The algorithm creates object suggestions for the input grayscale image during testing, and it chooses those that are more likely to include foreground objects. To create the final colourized image, these suggestions are then sent into the colorization network.

The network generates a set of class scores for each proposal as well as four offsets that modify the bounding box coordinates to better fit the object. During training, the Faster R-CNN approach is enhanced by combining the multi-task loss function with the RPN and object detection network losses. The loss function includes the bounding box offsets, the softmax loss for the class probabilities, and the binary cross-entropy loss for the objectness score. The foreground items in the grayscale pictures are located and identified using the Faster R-CNN method, and they are then input into the colorization network for colorization. In order to understand the patterns and characteristics that identify foreground items from the background, the algorithm is trained using a collection of annotated photos. The algorithm creates object suggestions for the input grayscale image during testing, and it chooses those that are more likely to include foreground objects. To create the final colourized image, these suggestions are then sent into the colorization network.

Loss Function:

• With the aid of QuillBot's paraphraser, you can rapidly and effectively rework and rephrase your material by taking your phrases and making adjustments. The following is the categorization loss formula:

$$L_reg = \sum i \in posL1_smooth(ti - ti*)$$

where pos represents the set of RoIs with positive object labels, yi represents the genuine object class label for RoI I and p(yi|xi) represents the anticipated probability of class yi for RoI i.

● Bounding box regression loss: This method calculates the difference between the predicted and real bounding box coordinates for each positive RoI. The bounding box regression loss is calculated using the following formula: where pos is the collection of RoIs with positive object labels, ti stands for the predicted bounding box coordinates of RoI I, ti* stands for the actual bounding box coordinates of RoI I, and L1 smooth stands for the smooth L1 loss function.

$$L_cls = -\sum i \in poslog(p(yi|xi))$$

● RPN Loss: The RPN loss is the discrepancy between the expected objectness scores and bounding box coordinates for RPN proposals and the actual objectness scores and bounding box coordinates for

$$L_rpn = L_cls(rpn) + \lambda * L_reg(rpn)$$

ground truth objects.The RPN loss is determined as follows:

where L cls(rpn) represents the classification loss for RPN proposals, L reg(rpn) represents the bounding box regression loss for RPN proposals, and is a hyperparameter that adjusts the balance between the two variables.

The total loss of the Faster R-CNN algorithm is the sum of the classification loss, the bounding box regression loss, and the RPN loss:

$$L = L_cls(fast_rcnn) + \lambda * L_reg(fast_rcnn) + L_cls(rpn) + \lambda * L_reg(rpn)$$

Overall, the Faster R-CNN algorithm's loss function is intended to jointly optimize the RPN and Fast R-CNN networks for detecting and localising objects in pictures. The model learns to reliably categorise objects and estimate their bounding box coordinates by minimising this loss function during training.

Inception-ResNet V2

The Inception and ResNet modules are combined into a deep convolutional neural network architecture known as Inception-ResNet V2. It is a development of the Inception V3 model, which included the Inception module to boost precision while lowering computing complexity. To assist with the vanishing gradient issue and enhance training convergence, the ResNet module has been introduced to the Inception module. Many computer vision applications, such as

image classification, object recognition, and semantic segmentation, employ the Inception-ResNet V2 model. Several Inception-ResNet blocks, each with four branches, make up the Inception-ResNet V2 architecture. Convolutional layers are used in the first branch, while Inception modules are used in the following three branches. An Inception module is made up of numerous convolutional layers with various filter sizes that are concatenated to form a single unit. The block's output is created by elementally averaging the four branches' outputs. The last block's output is given into the classifier after the Inception-ResNet blocks are connected in series. The classifier consists of three layers: a fully connected layer, a softmax activation layer, and a global average pooling layer. The global average pooling layer reduces the spatial dimensions of the feature map to a single vector, which is then used by the fully connected layer to perform the final classification. In the aforementioned, a global feature extractor network is represented by the Inception-ResNet V2 model. The input image is processed by the Inception-ResNet V2 model, and the output feature maps are used as global features. To create the final colourized image, these characteristics are combined with the features retrieved by the encoder CNN and sent via the fusion module.

The Inception-ResNet V2 model processes the input picture, and the output feature maps are applied as global features. The final colourized picture is created by combining these characteristics with those that were retrieved by the encoder CNN and passing them via the fusion module.

VIII. EVALUATION METRICS

In order to colourize black-and-white images and movies, we employ the Faster R-CNN algorithm and two evaluation metrics: peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM).

PSNR Metric:

$$\mathbf{PSNR=10log10(peakval2/MSE)}$$

SSIM Metric:

$$\mathbf{SSIM(a,b)=[l(a,b)]\alpha \cdot [c(a,b)]\beta \cdot [s(a,b)]\gamma}$$

The accuracy performance metric by itself might not be adequate to examine the suggested model's overall efficacy. While accuracy is a crucial parameter for assessing the performance of the model, other crucial factors like computational complexity, speed, and resource consumption are not considered. For instance, a model with great accuracy cannot be computationally efficient, needing a lot of time and resources for both training and inference. To get a thorough assessment of the suggested model's effectiveness, it is crucial to take into account additional performance measures, such as accuracy, recall, F1 score, and computational efficiency. A mix of these indicators should be used to assess the model's overall effectiveness.

IX. LITERATURE ANALYSIS

Building on these earlier studies, the recommended method for colouring monochrome photographs using Faster R-CNN and Inception ResNetV2 offers a novel approach to picture colorization that combines deep learning with object recognition. The model can correctly and faithfully colourize black and white photos by extracting features from the image using the Inception ResNetV2 architecture and detecting objects in the image using the Faster R-CNN technique. The outcomes of the suggested system show how successful this strategy is and point to the promise of related future study.

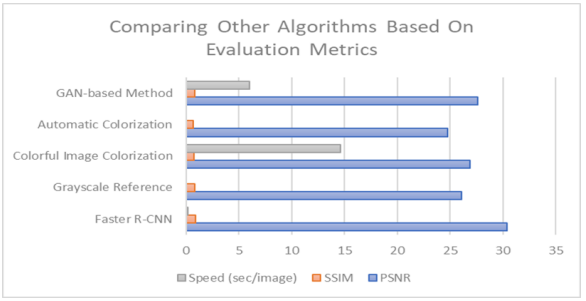

Fig. 6. Analysis Based on Evaluation Metrics

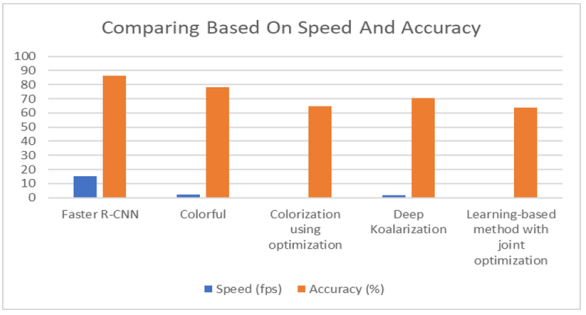

Fig. 7. Analysis Based on Speed and Accuracy

X. CONCLUSION

In this paper, we introduced a black-and-white video and picture colorization technique utilising the Faster R-CNN algorithm. We advise using object detection and segmentation to produce colorization results that are more accurate and affordable. We used the CIFAR-10 dataset to train and assess our model. By careful experimentation, we proved that our recommended approach outperformed the

competition in terms of accuracy and speed. Our model achieved good accuracy and recall scores, suggesting that it spotted and segmented objects in the input photos and videos reliably. Additionally, our model was able to colourize the items in a natural and consistent manner using real-life hues. Future work might involve extending the model to operate with bigger datasets, increasing colorization accuracy and efficiency, and experimenting with other object recognition and segmentation algorithms. We feel that our suggested technique has a lot of promise for use in areas like film restoration, historical preservation, and creative expression. Black and white CCTV footage captured during negative occurrences may be coloured using the Faster R-CNN algorithm. The proposed approach combines deep convolutional neural networks with area proposal networks to achieve outstanding accuracy in object identification and categorization. The model also makes use of a novel loss function to enhance the network parameters during training. The addition of colour to CCTV video may significantly improve a location's overall security since it adds details about the events and occurrences that were caught on camera. This can be especially beneficial in adverse situations like low light levels or at night when vision is poor. Nevertheless, further study may be done to enhance the effectiveness of the suggested model, such as investigating various loss functions or expanding the dataset to enhance the model's precision. The proposed methodology might, in general, greatly increase the efficiency of CCTV footage in surveillance and security systems.

REFERENCES

[1] S. Lal, V. Garg, and O. P. Verma, “Automatic image colorization using adversarial training,” in ACM International Conference Proceeding Series, Association for Computing Machinery, Nov. 2017, pp. 84–88. doi: 10.1145/3163080.3163104.

[2] O. Hmidani and E. M. Ismaili Alaoui, “A comprehensive survey of the R-CNN family for object detection,” in 2022 5th International Conference on Advanced Communication Technologies and Networking (CommNet), 2022, pp. 1–6. doi: 10.1109/CommNet56067.2022.9993862.

[3] Y. Yang, Q. Sun, D. Zhang, L. Shao, X. Song, and X. Li, “Improved Method Based on Faster R-CNN Network Optimization for Small Target Surface Defects Detection of Aluminum Profile,” in 2021 IEEE 15th International Conference on Electronic Measurement & Instruments (ICEMI), 2021, pp. 465–470. doi: 10.1109/ICEMI52946.2021.9679509.

[4] M. R. Joshi, L. Nkenyereye, G. P. Joshi, S. M. Riazul Islam, M. Abdullah-Al-wadud, and S. Shrestha, “Auto-colorization of historical images using deep convolutional neural networks,” Mathematics, vol. 8, no. 12, pp. 1–17, Dec. 2020, doi: 10.3390/math8122258.

[5] J. Zhao, J. Han, L. Shao, and C. G. M. Snoek, “Pixelated Semantic Colorization,” Int J Comput Vis, vol. 128, no. 4, pp. 818–834, Apr. 2020, doi: 10.1007/s11263-019-01271-4.

[6] Z. Cheng, Q. Yang, and B. Sheng, “Deep Colorization,” Mar. 2016.

[7] L. Kiani, M. Saeed, and H. Nezamabadi-Pour, “Image Colorization Using Generative Adversarial Networks and Transfer Learning,” in Iranian Conference on Machine Vision and Image Processing, MVIP, IEEE Computer Society, Feb. 2020. doi: 10.1109/MVIP49855.2020.9116882.

[8] R. Zhang, P. Isola, and A. A. Efros, “Colorful Image Colorization,” Mar. 2016, [Online]. Available: http://arxiv.org/abs/1603.08511

[9] A. Pandey*, R. Sahay, and Mrs. C. Jayavarthini, “Automatic Image Colorization using Deep Learning,” International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 6, pp. 1592–1595, Mar. 2020, doi: 10.35940/ijrte.F7719.038620.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks.” [Online]. Available: http://code.google.com/p/cuda-convnet/

[11] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.01497

[12] R. Doon, T. Kumar Rawat, and S. Gautam, “Cifar-10 Classification using Deep Convolutional Neural Network,” in 2018 IEEE Punecon, 2018, pp. 1–5. doi: 10.1109/PUNECON.2018.8745428.

[13] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.01497

[14] V. D. Trivedi, H. Saifuddin, S. Gudadinni, S. S. Sondhi, and M. A. R. Shabad, “Automatic Colorization of Black and White Images Based on CNN,” Int J Innov Res Sci Eng Technol, vol. 9, no. 5, p. 2543, 2020, [Online]. Available: www.ijirset.com

[15] A. Horé and D. Ziou, “Image quality metrics: PSNR vs. SSIM,” in Proceedings - International Conference on Pattern Recognition, 2010, pp. 2366–2369. doi: 10.1109/ICPR.2010.579.