DeepColor: Enhancing Image Colorization with Advanced Deep Learning Techniques

Colorize with Deep Learning: From Grayscale to Vibrant Hues!

Anna Tsvetkov (atsvetko), Samuel Walhout (swalhout), Taojie Wang (twang49)

Source Code: https://github.com/annatsv/deepcolor

I. Introduction

How do we colorize images using AI? What are the technical and ethical challenges?

DeepColor is driven by new advancements in AI colorization. Recently, there has been a surge of interest in AI-driven colorization both in popular media and academic literature. We take on the problem of colorizing images using advanced deep learning techniques. On the technical side, we explore how well Convolutional Neural Networks (CNNs) can understand and replicate colors and their complex distributions in images. On the ethical side, we examine the implications of using AI to modify images. We are motivated by the idea that advancing the capabilities of neural networks in image colorization can contribute to the "big-picture" question of how AI can understand and replicate human perception. This has important practical applications in the remastering of historical images, visual art, and surveillance, and also deepens our understanding of the perceptual processing of AI.

We build upon an existing paper "Deep Koalarization: Image Colorization using CNNs and Inception-ResNet-V2" by Frederico Baldassarre, Diego González Marín, and Lucas Rodés-Guirao. The paper introduces a model that combines a CNN trained from scratch with high-level features that were extracted from the pre-trained Inception-ResNet-v2 model for image colorization. We chose this paper because it presents a novel approach to image colorization that we believe can be streamlined and improved upon with advanced deep learning techniques and architectures.

Our work innovates on the original paper by employing different architectures, datasets, and hyperparameters, including U-Net, DenseNet and MobileNet, to evaluate the impact on colorization accuracy. We streamline certain features of the original model and test several advanced architectures. The upshot of our project **DeepColor** is more accurate and visually appealing colorized images.

II. Methodology

Data

We sampled images from three prominent datasets for our project. Importantly, these are different from the data used in the original paper we are replicating, which used a subset of 60,000 images from the ImageNet dataset for their model. Due to limitations in our computing resources, processing and colorizing such a large number of images was not feasible. Handling large datasets requires a significant amount of memory and computational power, which exceeds the capacity of our available hardware. To address this, we experimented with different batch sizes of images (e.g. 100, 200, 300, 400) to find a balance between computational efficiency and the performance of our model and settled on a batch size of 383 and 15,000 total images.

First, we used a subset of 5000 images from the **COCO Dataset** (Common Objects in Context). We chose this dataset because it provides a diverse range of images with different scenes and objects which will help our model learn to colorize a wide range of subjects and scenarios. The COCO dataset contains over 200,000 labeled images, including complex scenes with multiple objects.

Second, we used a subset of 5000 images from the **Places365 Dataset**. This dataset is specifically designed for scene recognition and has over 2.5 million images covering more than 205 unique scene categories. By including images from Places365 in our training data, we aimed to enhance our model's ability to colorize various environments, such as natural landscapes to urban settings. This complements the object-centric focus of the COCO dataset and provides comprehensive learning for our models.

Third, we used a subset of 5000 images from the **ADE20K Dataset** which contains more than 20,000 images annotated with objects and object parts. This dataset provides our model with the opportunity to learn the colorization of both objects and their parts in detail, which is important for achieving realistic colorization results.

Each subset of 5000 images was split into training and validation sets with 4000 images allocated for training and 1000 images for validation for each dataset for each model. So in total each model had 12000 images in training and 3000 images in validation. We processed all of the data in four main preprocessing steps:

1. **GreyScale Conversion:** We converted the color images to grayscale to create the input for our colorization model. This step is important as our models aim to

learn how to add color to grayscale images. It's also important to note that we chose to use colored datasets (and not grayscale datasets) to retain the original color images as the "ground truth" for training and evaluating our model. This allows us to compare the colorized output of our model to the original color images and calculate metrics such as Mean Squared Error (MSE) or Peak Signal-to-Noise Ratio (PSNR) to assess the colorization accuracy.

2. **Resizing Images from the Datasets:** We resized the images to a uniform size (e.g., 256 x 256 pixels) to ensure consistency across the datasets and reduce computational load during training. We made this choice based on the computational resources available, the desired level of detail in the selected colorized images, and uniform batch processing.

3. **Normalization**: We normalized the pixel values to a range that is suitable for the input to our network [0, 1] to help with our training process and to improve convergence. This helps us make sure that all input features (pixel values in our case) are treated equally by the models.

4. **Data Augmentation**: We tested the models with data augmentation techniques such as rotation and flipping. The purpose of this was to help our model generalize to unseen data by simulating variations that might occur in real-world scenarios (e.g. objects getting rotated, flipped over, etc.). *But* we found that these techniques distorted the images in ways that adversely affected the training results. So we eliminated this data augmentation step to better align the training process with the original paper and our goal of maintaining the integrity of the image and colorization accuracy.

Figure 1. This figure illustrates the preprocessing steps in our study and shows the transformation of various raw images into their preprocessed (greyscale, resized and normalized) versions.

Models

The original paper proposed a model that combines a deep Convolutional Neural Network (CNN) trained from scratch with high-level features extracted from the pre-trained Inception-ResNet-v2 model. Their architecture includes an encoder-decoder structure with a fusion layer that integrates the features from the Inception-ResNet-v2 model. This fusion layer basically makes sure that the semantic information provided by the Inception-ResNet-v2 model is distributed across the spatial regions of the image, which helps the decoder in generating a more accurate and detailed colorization.

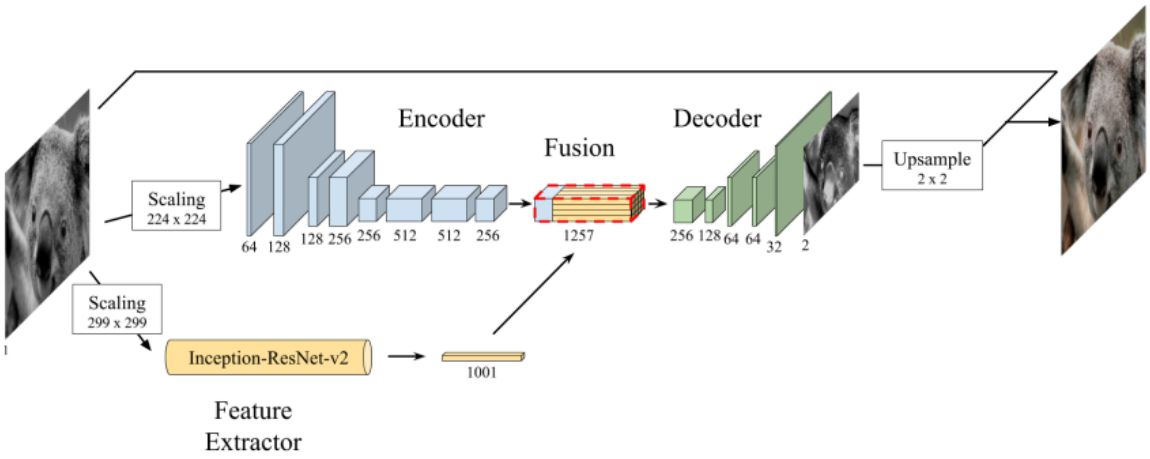

Figure 2. Model architecture from the "Deep Koalarization" paper. In place of the original fusion layer we employ advanced architectures U-Net, DenseNet, or MobileNet to evaluate performance, potentially eliminating the need for additional fusion layers and streamlining the CNN-based colorization task.

In **DeepColor** we experimented with the following architectures, and include our rationale for selecting each one below:

1. **U-Net:** U-Net is effective in image segmentation tasks. U-Net uses skip connections that can help us retain important spatial information and reconstruct the detailed color in the images. We hypothesized that U-Net's ability to capture both local and global features might be beneficial for image colorization.

2. **DenseNet:** DenseNet has dense connections between layers, which can enhance feature propagation and reduce the number of parameters in our model. We believed that DenseNet's architecture could provide a good balance between the complexity of our model and its colorization quality.

3. **MobileNet:** MobileNet has compact and layer-specific connections which can potentially reduce computational cost while maintaining accuracy. This is important for our work given our limited computational resources. We experimented with MobileNet to explore the trade-off between model size and colorization performance.

Similarly to the original paper, we used the Adam optimizer with a learning rate of .0001. We used a fixed image size and a batch size of 383 images per epoch. For our models (U-Net, DenseNet, and MobileNet), we trained them with the objective loss function being the Mean Squared Error (MSE) between the estimated colors of the pixels and their real-world values, derived from the ground-truth, unpreprocessed, full color images. All models return output image shapes of (256,256,3).

The U-Net model was trained without a pre-trained base model. It consists of four encoder blocks which convolve over the input, apply batch normalization, and activate the output using the LeakyRelu function with an alpha of .01. Next, there are three decoder blocks which perform a transpose convolution, integrate a skip connection, then apply normalization, dropout, and activation layers. A dropout rate of 0.5 and the ReLU function were used. A final transpose convolution is performed with a sigmoid activation function to produce the final output.

The DenseNet model was trained using the pre-trained DenseNet121 as a base model. The weights were pre-trained on the Image Net dataset. The inputs are fed into the base model which acts as the encoder of the model. The encoded results are passed through 5 decoder blocks. Each block consists of an upsampling layer and a 2D convolution. Then, an activation function is applied along with a batch normalization layer. The first four blocks use a ReLU activation function. The last block uses a sigmoid activation function and forgoes the batch normalization layer.

The MobileNet model is similar to the DenseNet model, except that MobileNet uses MobileNet as a base model for the encoder, also pre-trained on the image net dataset. Our MobileNet model then goes through 5 cycles of upsampling, convolving, and activating with the ReLU function. There is then a final sixth convolution with a sigmoid activation function.

Metrics

Success is primarily measured by the accuracy of colorization by our model. We assess the success of our colorizers hrough quantitative and qualitative analysis. In addition to accuracy, we will also consider the visual appeal and the naturalness of the colorized

images as a metric for success since our ultimate goal is to produce images that are not only accurate but also aesthetically pleasing to perceivers.

We believe that these new metrics improve upon the original paper. The authors of the "Deep Koalarization" paper we are building upon aimed to show that a model that combined a CNN with high-level feature extraction from the Inception-ResNet-v2 model could accurately colorize images. *However* they primarily quantified their results through qualitative assessments. For their quantitative assessments, they mentioned using Mean Squared Error (MSE) as the objective function during training but did not provide explicit MSE values as a measure of colorization accuracy in the results section.

Instead, they conducted a user study to assess the "public acceptance" of the colorized images, where participants were asked to judge whether the colorized images looked real or not. In the paper, "public acceptance" refers to the percentage of participants in a user study who mistakenly identified colorized images as real color images. A higher rate of public acceptance indicates that the colorized images are more convincing and indistinguishable from true color images to human observers. This qualitative approach allowed them to gauge both the perceived realism of their model's output and its visual appeal, which is what we are also trying to achieve.

In **DeepColor**, we use both qualitative and quantitative assessments, including a user study as well as quantitative assessments like Mean Squared Error (MSE) or Peak Signal-to-Noise Ratio (PSNR) between the colorized and original images.

Results

Our *base goal* was to achieve colorized outputs for each model and approximate the performance of the original paper in terms of colorization quality. We were aiming for a similar rate of "public acceptance" in a user study, which was reported to be 45.87% in the original paper, and use additional metrics such as Mean Squared Error (MSE) or Peak Signal-to-Noise Ratio (PSNR) between the colorized and original images, even though the original paper did not provide explicit quantitative metrics.

Our *target goal* was to improve upon our quantitative and qualitative assessments by experimenting with different architectures and hyperparameters. This could involve achieving higher "public acceptance" rates in a user study (e.g., exceeding 45.87%) or good scores on quantitative metrics like MSE or PSNR.

Our *stretch goal* was to achieve state-of-the-art colorization accuracy that potentially surpasses the model in the original paper as well as other existing models in the field of colorization. This would involve significantly higher "public acceptance" rates and

superior quantitative metric scores, indicating that our model can produce highly realistic and accurate colorizations. Here are the qualitative results of our three models:

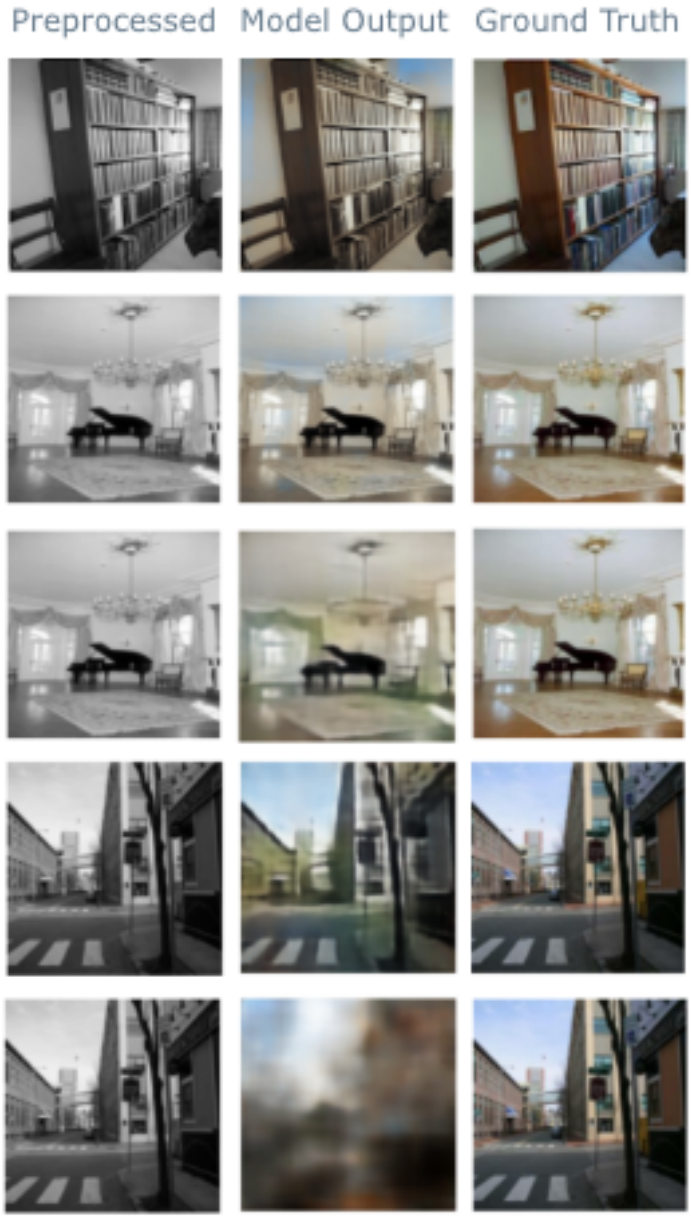

Figure 3: Comparison of Colorization. First two rows are U-Net, the next two are DenseNet, and the final row is MobileNet. Notably, all models achieved colorization though with varying degrees of accuracy.

We conducted a public acceptance study and used MSE and PSNR as additional metrics. We also qualitatively evaluated the images. Each of the models performed differently in reference to the goals set at the beginning of the project. Here are the visualized results of our user study:

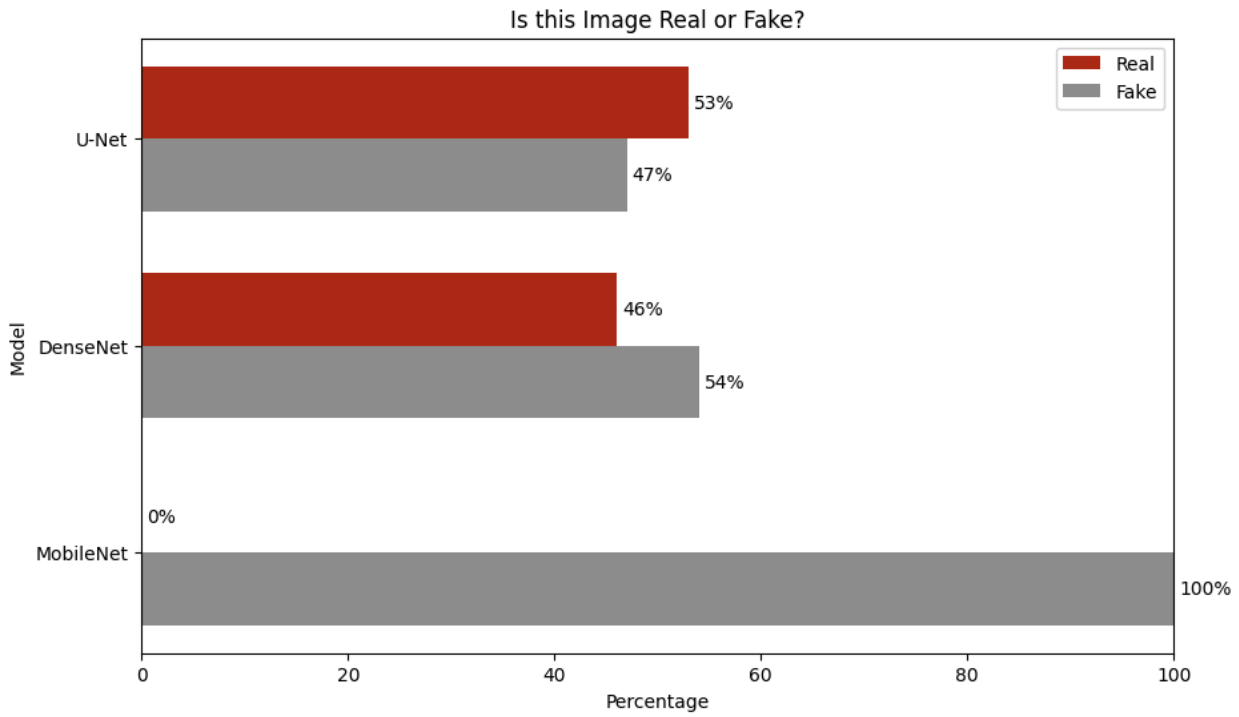

Figure 4. This figure illustrates the public acceptance of our generated colorized images. The sample size was 30 undergraduate and graduate students at Brown.

U-Net met the base goal and the target goal. It achieved a user acceptance of 53% which marks a substantial improvement over the 45.87% acceptance rate in the original paper. The Mean Squared Error (MSE) for the U-Net model was 0.006. The Peak Signal-to-Noise Ratio (PSNR) for the U-Net model was 23.799. The MSE and PSNR cannot help us compare our models to the original study, but they are useful for comparing the models between each other. Qualitatively, U-Net produced sharp, clear images with generally correct colors. The predicted colorizations were often less saturated than the original photos but still accurate. See Figure 5 for the training and validation dynamics for U-Net.

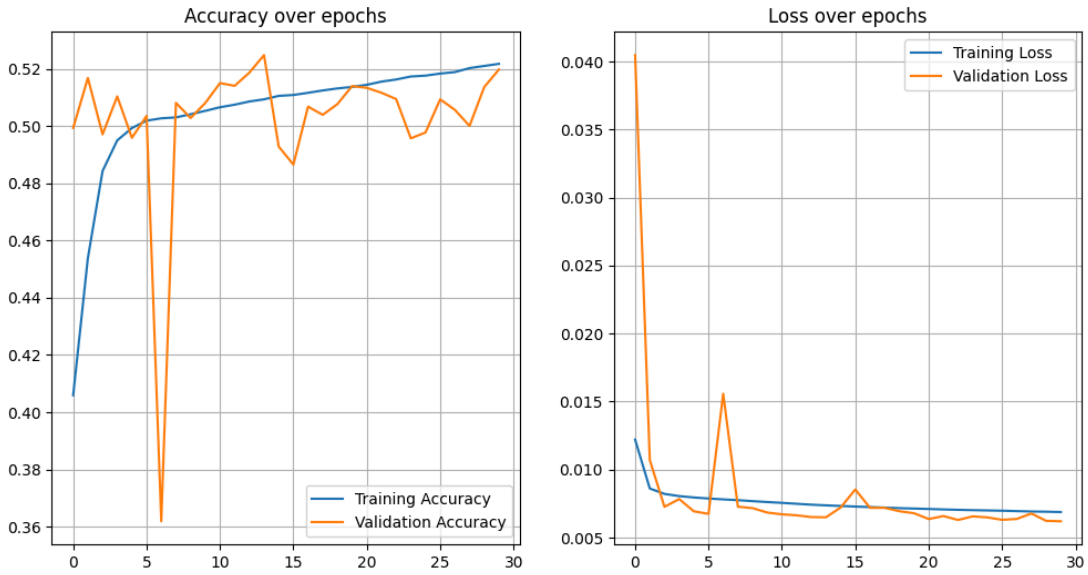

Figure 5. Example Training and Validation for U-Net. U-Net accuracy and loss over 30 epochs shows rapid early improvement and stabilization.

DenseNet also met the base goal and target goal, achieving a 46% acceptance rate. This is a marginal improvement in acceptance compared to the original paper. The Mean Squared Error (MSE) for the DenseNet model was 0.013 which is about double the MSE of U-Net. The Peak Signal-to-Noise Ratio (PSNR) for the DenseNet model was 19.469 which is slightly lower than U-Net. Qualitatively, DenseNet produced slightly fuzzier images than U-Net. It displayed a strong ability to correctly predict colors, but less ability to preserve sharp features and details in the predicted colorizations. See Figure 6 for the training and validation dynamics for DenseNet.

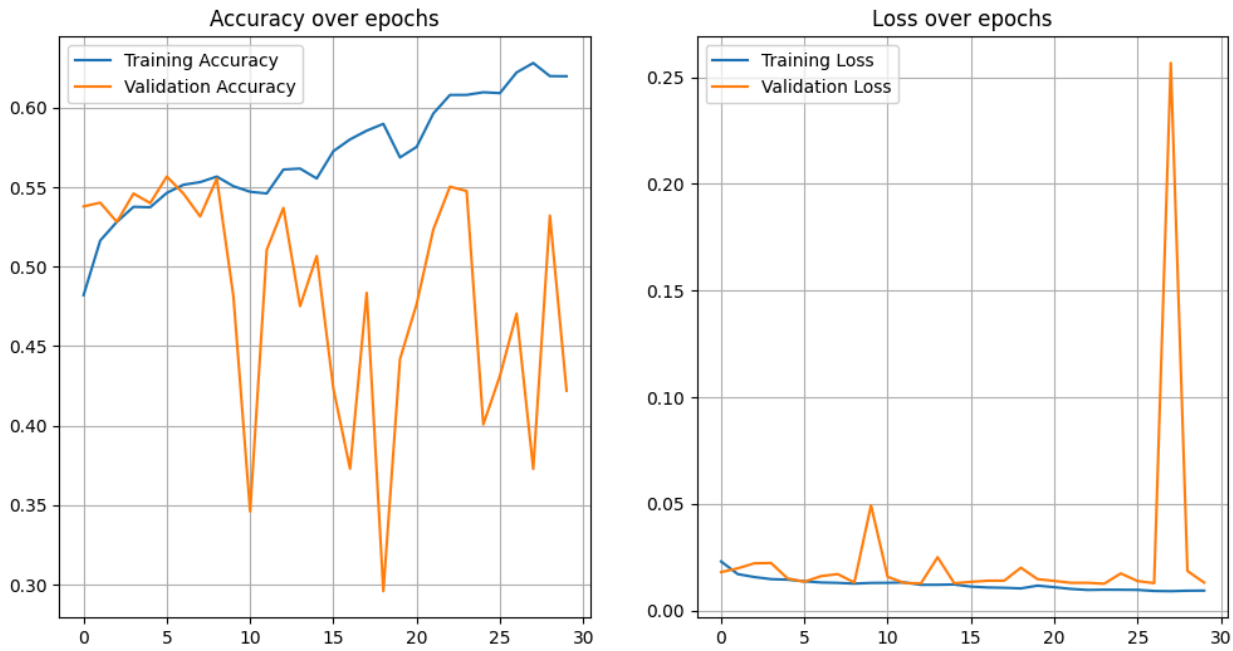

Figure 6. Example Training and Validation for DenseNet. DenseNet accuracy and loss over 30 epochs shows a steady increase in training accuracy while validation accuracy fluctuates significantly. It also shows that both training and validation loss decrease over time with a spike in validation loss during the final epochs.

MobileNet did not meet the base goal. While it did achieve colorized outputs, and so met some expectations, its colorizations were not accepted by the general public. The Mean Squared Error (MSE) for the MobileNet model was 0.053 which is almost five times larger than DenseNet and almost ten times larger than U-Net. The Peak Signal-to-Noise Ratio (PSNR) for the MobileNet model was 13.102 which is much lower than both U-Net and DenseNet. Qualitatively, MobileNet produced colorization predictions that did not match the ground truth images. MobileNet's colorizations can be characterized as extremely noisy blends of color without the definitive features of the original pictures represented. While the correct colors were often located in the correct general areas of the image, ultimately, they lacked any real coherence. As we reflect on our project, this was perhaps not surprising and due to the lightweight architecture of MobileNet which struggles to capture the finer details necessary for accurate colorization. See Figure 6 for the training and validation dynamics for MobileNet.

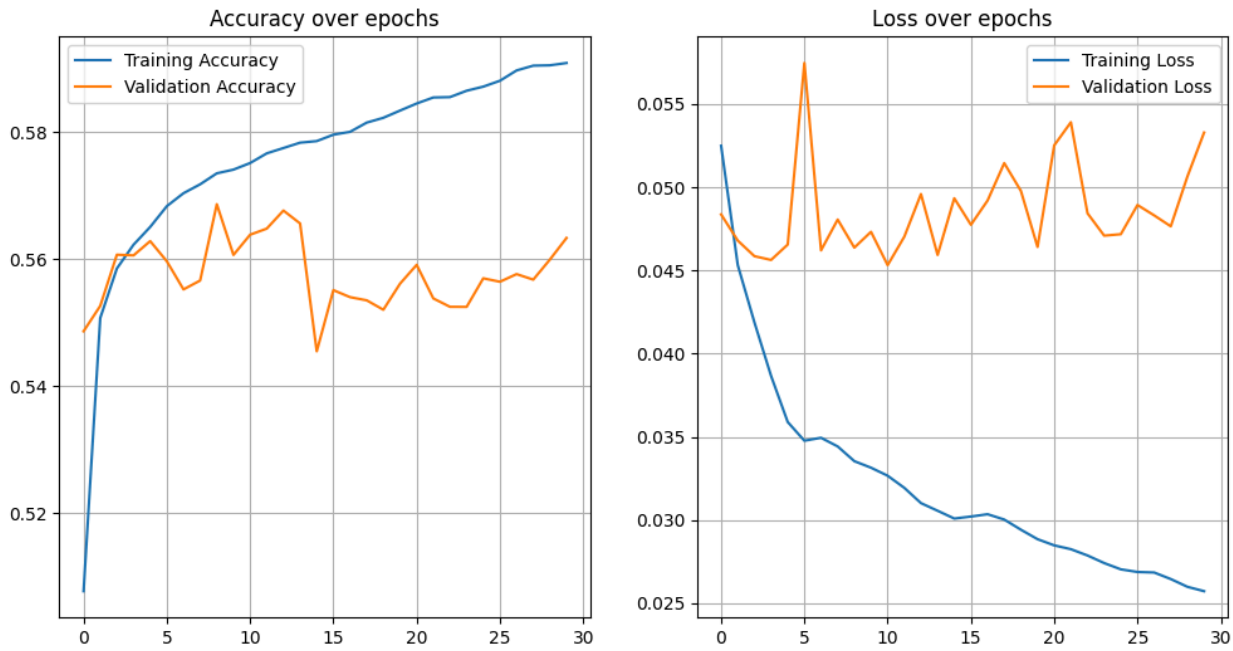

Figure 7. Example Training and Validation for MobileNet . MobileNet accuracy and loss over 30 epochs shows that the training accuracy of the model increases over 30 epochs and the training loss consistently decreases. However, the validation accuracy shows less change and the validation loss is volatile with spikes at epochs 10 and 25. These signs indicate that the model might not perform well on new, unseen data.

We conducted an ablation study on the number of epochs to determine the optimal tradeoff between computing resources and model accuracy. Our results from above were produced by models trained for 30 epochs. We can see based on the data from training the models on 10 epochs instead of 30 epochs, the models have similar levels of performance. This indicates that future models designed to colorize images likely do not need to train for an excessive number of epochs. See Figure 8 for a qualitative comparison between model outputs for U-Net for 10 and 30 epochs. And See Figure 9 for a quantitative assessment of different metric evaluations with respect to different numbers of epochs for all models.

Figure 8. Visualizations of model outputs for U-Net above. First 5 rows are colorized images produced at 10 epochs. Final 5 rows are colorized images produced at 30 epochs.

Table 1: Colorization Performance Across Different CNN Architectures

Model	MSE (10 Epochs)	PSNR (10 Epochs)	MSE (30 Epochs)	PSNR (30 Epochs)
U-Net*	0.006	23.229	0.006	23.799
DenseNet	0.014	19.073	0.013	19.468
MobileNet	0.049	13.366	0.053	13.102

Figure 9: Quantitative Performance Metrics. A higher Mean Squared Error (MSE) indicates poorer image quality as it measures the average squared intensity differences between the ground truth and colorized image. A lower Peak Signal-to-Noise Ratio (PSNR) indicates poorer image quality as it decreases logarithmically with an increase in MSE score and signifies that there is greater distortion in the colorized image. *U-Net demonstrates the best performance with minimal MSE and maximal PSNR. MobileNet shows consistent low performance. Extended training moderately benefits U-Net and DenseNet.

III. Challenges

There were four large challenges in the project that were all resolved. One challenging aspect of our project was dealing with the architectural complexities of the three models we implemented. In particular, the upsampling processes required to match the output image size with the input dimensions was challenging. To work through this, we experimented with different configurations of upsampling layers. This process required extensive testing and validation to ensure that the network architecture was configured correctly to upscale without losing detail and to match the shape of the images.

A second challenge was managing the huge sizes of our chosen datasets and making subsets for training proved to be challenging due to our limited computational resources. We worked through this by using really efficient data management libraries and platforms. Specifically, we leveraged TensorFlow datasets and Hugging Face's datasets library for the Places and ADE20k datasets, which provide streamlined access to subsets of the datasets, which really reduced our load time and memory overhead. For the Coco database, we used the FiftyOne library, which is particularly useful for visualizing and filtering the large dataset efficiently. These tools allowed us to handle large volumes of data more effectively, enabling us to focus on model training and optimization without being hindered by our hardware limitations.

A third challenge was the extensive amount of time and computing resources required to train the models. In order to train the model on time, two group members bought Google Colab Pro to gain access to additional cloud computing resources including GPUs, and even with these resources the models each took hours to train (sometimes running overnight for up to 7-8 hours each) before extensive optimization. Our CNN architectures are extremely computationally expensive especially without access to

GPUs which can parallelize computations and speed up the process immensely. Due to computation restrictions, the scale of our ablation studies had to be cut back. The original plan involved more rigorous testing and architecture specific hyperparameter tuning, including but not limited to testing the efficacy of skip connections within U-Net and the benefits of a pre-trained base model within DenseNet and MobileNet. Nevertheless we still implemented informative ablation studies with respect to our layer configurations and the number of epochs in training, as evident in Figure 8.

A fourth challenge was the loss of our fourth group member. When we originally proposed this project, we were expecting to have an additional member to help share the workload. Since we have not chosen to pare down the scope of our project, it has resulted in additional responsibilities for each group member.

IV. Reflection

Our project **DeepColor** turned out to be a success. Two of the three architectures were able to hit the target goal of improving on the original paper's acceptance rate and all three architectures were able to produce colorized output. Moreover, as evident in Figure 3 in the qualitative results of our best model, while some of the images certainly could look closer to the ground truth, an impressive amount of detail and color is accurately predicted.

In the middle of the project that was not the case. The first time we trained the models the results were not promising. U-Net produced black squares, DenseNet produced random fuzzy blobs, and MobileNet merely returned the black and white image. We were able to develop our models and fix our preprocessing and technical pipelines to get all of the models predicting colorizations that, at bare minimum, resemble the ground truth and in many cases closely mimic the ground truth. See Figure 10 for preliminary results before extensive refinement of our technical pipeline and before we resolved all these issues to achieve accurate and visually pleasing colorized results.

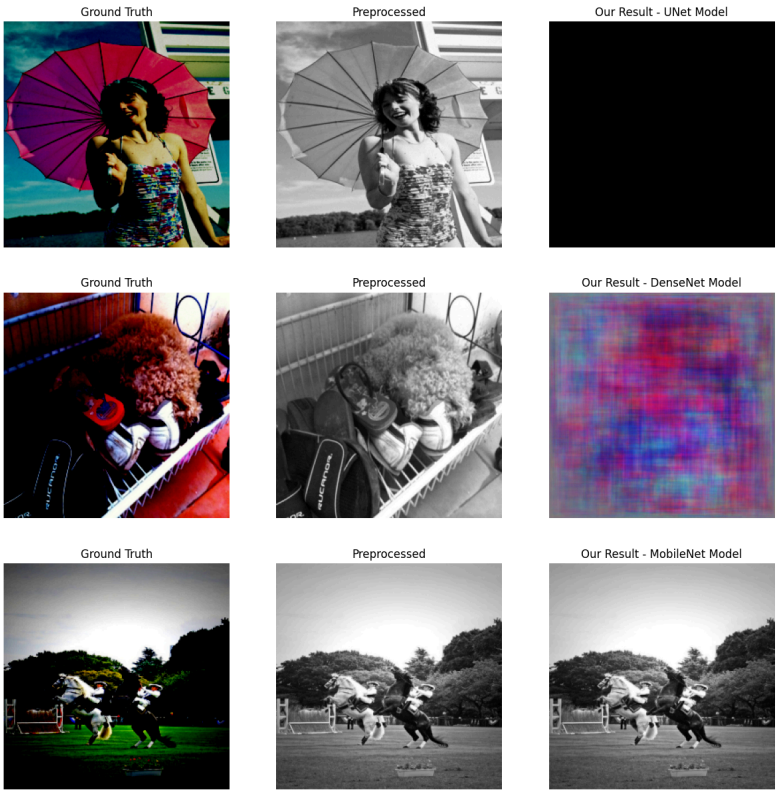

Figure 10. Initial results early on in our experiments before extensive debugging and refinement of the technical pipeline. As compared to the final and *much more* accurate results from our models seen in Figure 3 we have come a long way!

We did not expect U-Net to be the most accurate architecture. DenseNet and MobileNet both use a pre-trained base model within their architecture. Therefore, we expected the highest performing model to be either DenseNet or MobileNet. U-Net in comparison had fewer layers and was completely coded by us. This tells us that the skip connections are extremely important for colorization tasks and we learned that a pre-trained feature extraction model is not necessary to achieve a high degree of accuracy.

We had to change course when we encountered the limitations of our computational resources. We had originally planned a more robust set of ablation studies that we were forced to limit. If we were to redo this project, we would spend more time making sure we have the requisite resources to complete training in a practical manner. We might also consider how we could leverage Brown Computer Science's existing computational resources to our advantage. Another iteration of this project attempting once again to colorize black and white images would be interesting to implement. A model that uses the best parts of U-Net and DenseNet might be even more effective than the models we were able to build in this project. More experimentation with hyperparameters could help the model to optimize performance. Using different datasets could challenge our model and find blindspots we did not account for.

DeepColor helped us go beyond the theory of deep learning into the practical challenges and applications of the technology we have been learning about in lecture. We learned how to source and preprocess data. We created models from scratch and evaluated the tradeoffs in different architectural decisions. We developed the skills necessary to compile results into a digestible format for anyone to understand. Overall, we learned how to take a simple problem, research state of the art techniques, and apply them in a deep learning framework to find a solution. This cycle of research, application and experimentation is foundational to our work as computer scientists.