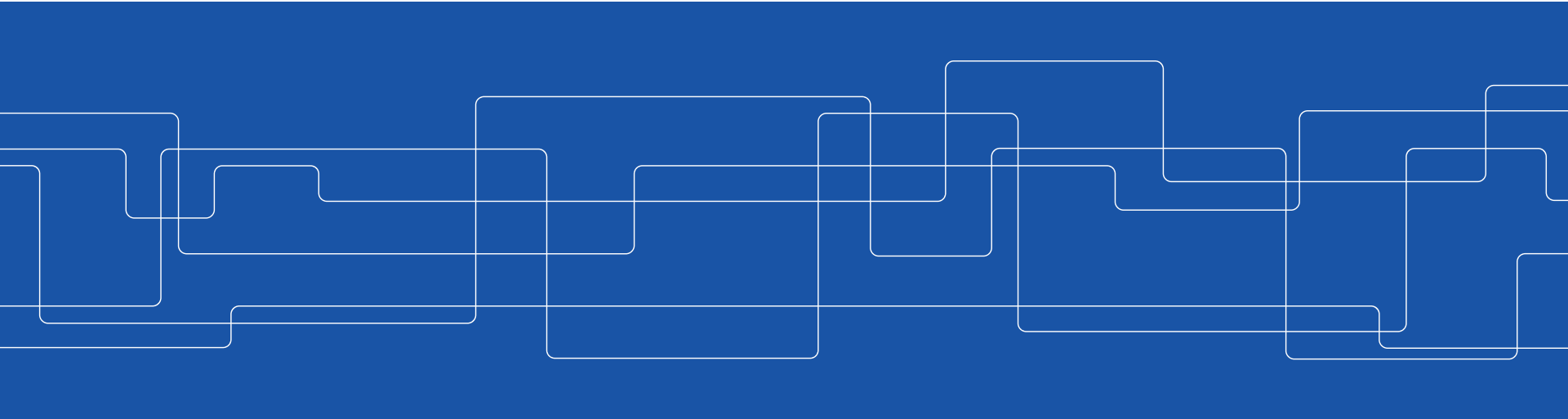
Deep Image Colorization

GROUP 72

Fadhil Mochammad
fadhilmch@kth.se

Muhammad Irfan Handarbeni
handa@kth.se

Sri Datta Budaraju
budaraju@kth.se



Aim

Colorization of Gray Scale Image

Applications :

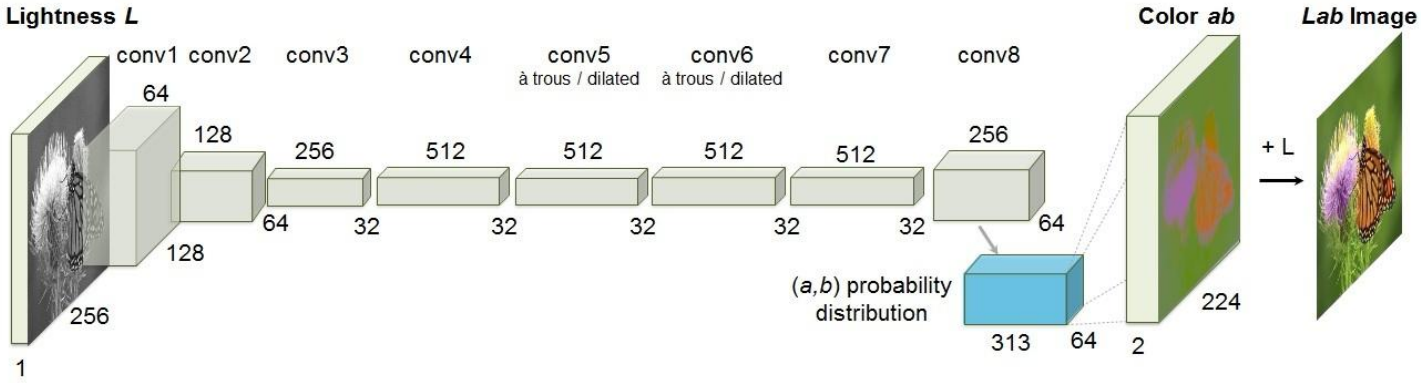
- Remastering old pictures
- Enhancing CCTV footages for crime fighting

Motivation

To explore multiple concepts in Computer Vision

- Convolutions
- Autoencoder
- Pre-Trained Model
- *GANs

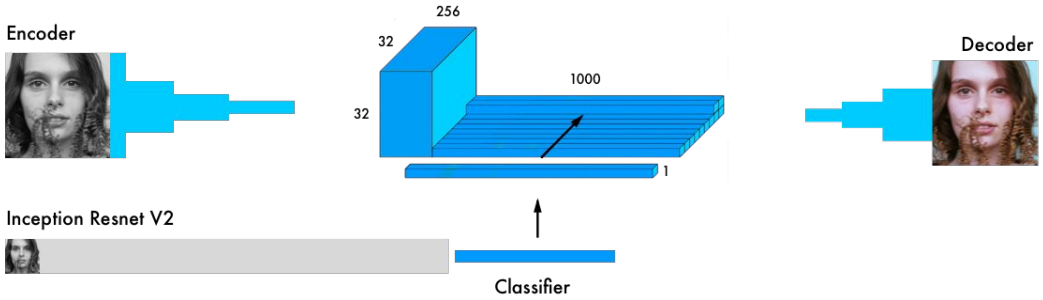
Inspirations - Cool - Colorful Image Colorization



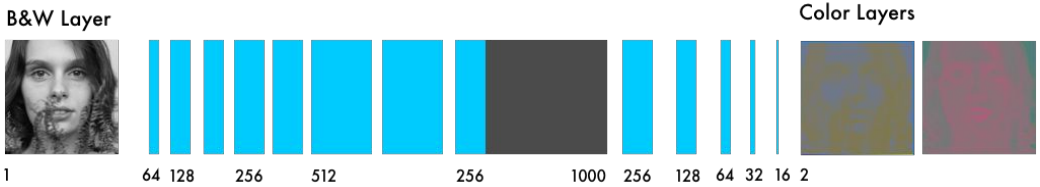
Probabilistic predictions of color.
Color specific Loss

Source：https://arxiv.org/pdf/1603.08511.pdf

Super cool - Colorizing B&W Photos with Neural Networks / Deep Koalarization

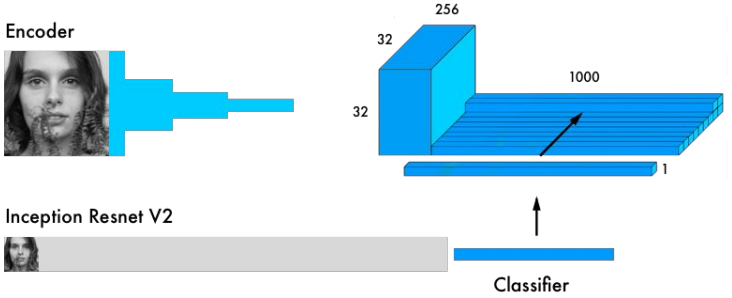


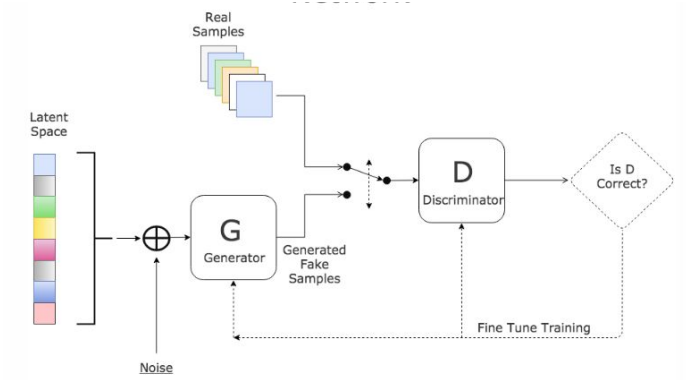
Leveraging Pre-Trained Models



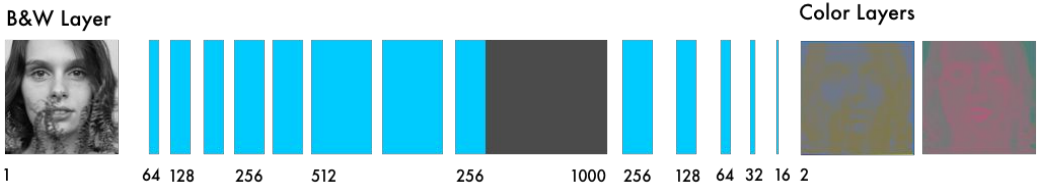
Source：https://mc.ai/understanding-gans-with-deoldify/

Super Super cool Deoldify





GANS!



Source：https://mc.ai/understanding-gans-with-deoldify/

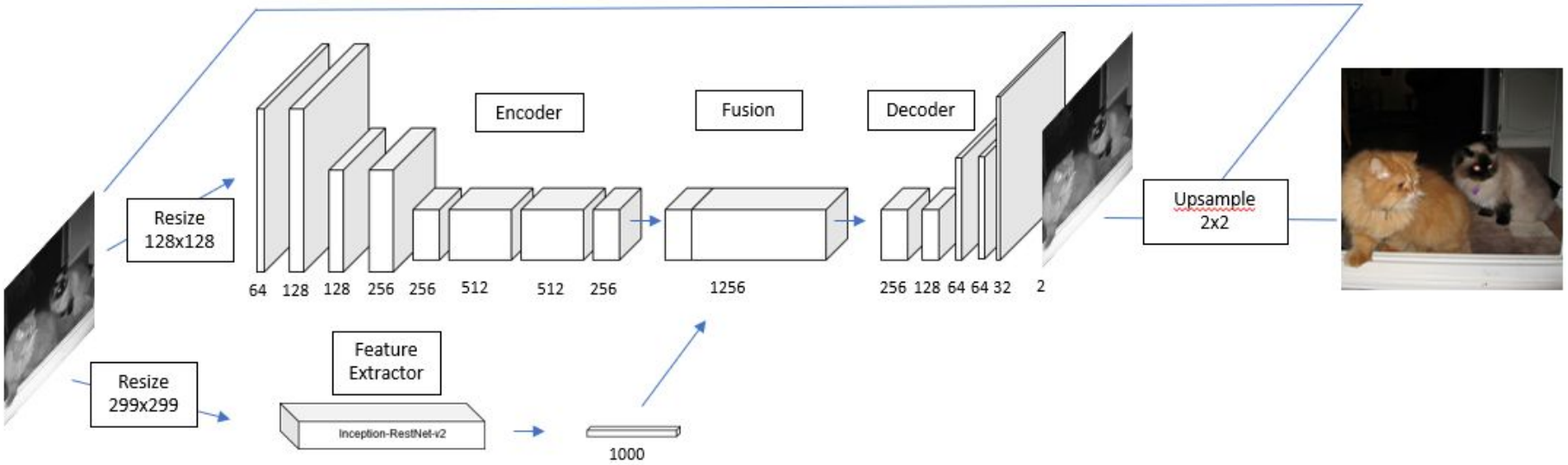
Approach

- Represent Images into CIE L*a*b* color space
  - L* (grayscale) channel represent the image lightness
  - a* and b* channels represents spectra green-red and blue-yellow respectively
- The L* channel is used as the input for our model and the model will estimate the a* and b* channels

Approach

- Use CNN-based approach which is widely used in Computer Vision
- Use autoencoder concept in the colorization pipeline, where the first half is the encoder which extract the important features of the image, and the second half is the decoder to reproduce the image from the extracted features
- Make use of pre-trained Inception-ResNet-v2 to get high-level features of the image
- The estimated a* and b* channels are combined with original grayscale input to get colorized image

Architecture



Dataset

- Tiny Image Net
- ImageNet

Experiment

Experiments to evaluate the capabilities of the model over different volumes of training data:

- Experiment on Tiny-Imagenet Data
- Class overfit experiment on Imagenet Data
- Multiclass experiment on Imagenet Data
    - Precompute pre-trained embedding features and save data into TFRecords to cut training time
- Colorizing Old Photo

Training

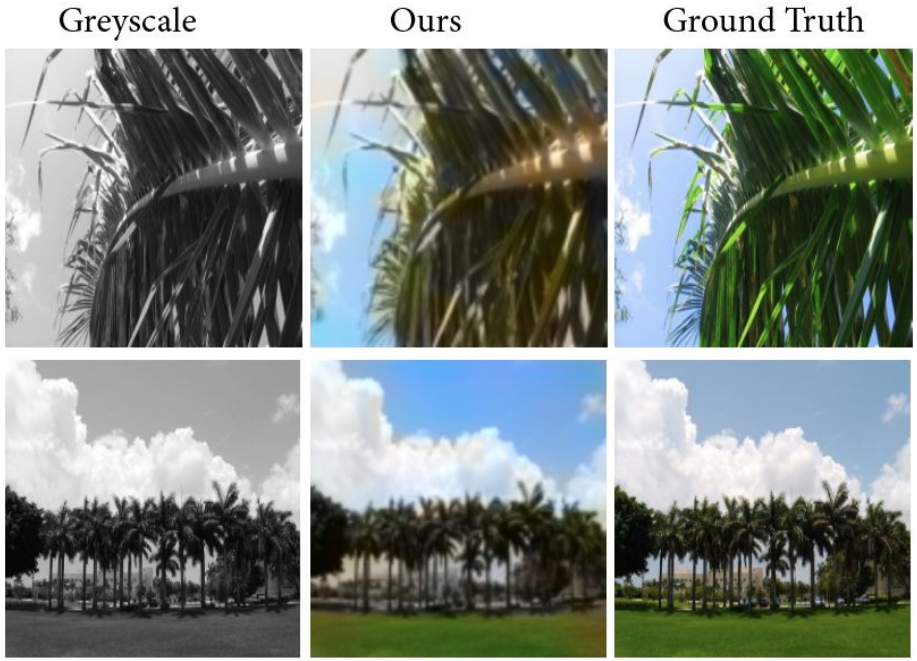
- Save model when there is improvement on score
- Reduce Learning rate by a factor of 0.1 if no improvements for consecutive few epoch.
- Stop training if we are not seeing any improvement at all during several training.

Experiments - 300 Images Tiny-Imagenet (Class Overfit)

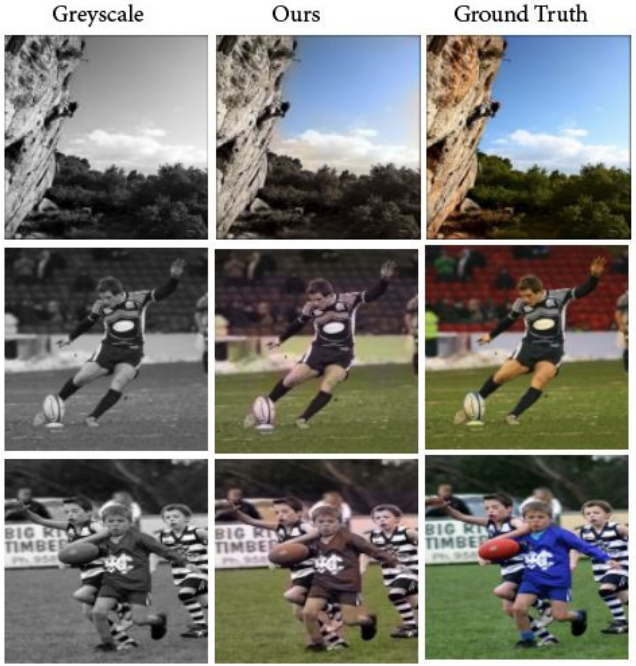
64x64 pixels

Experiments - 200 Images (Class Overfit)

Experiments - 1000 Images (Class Overfit)



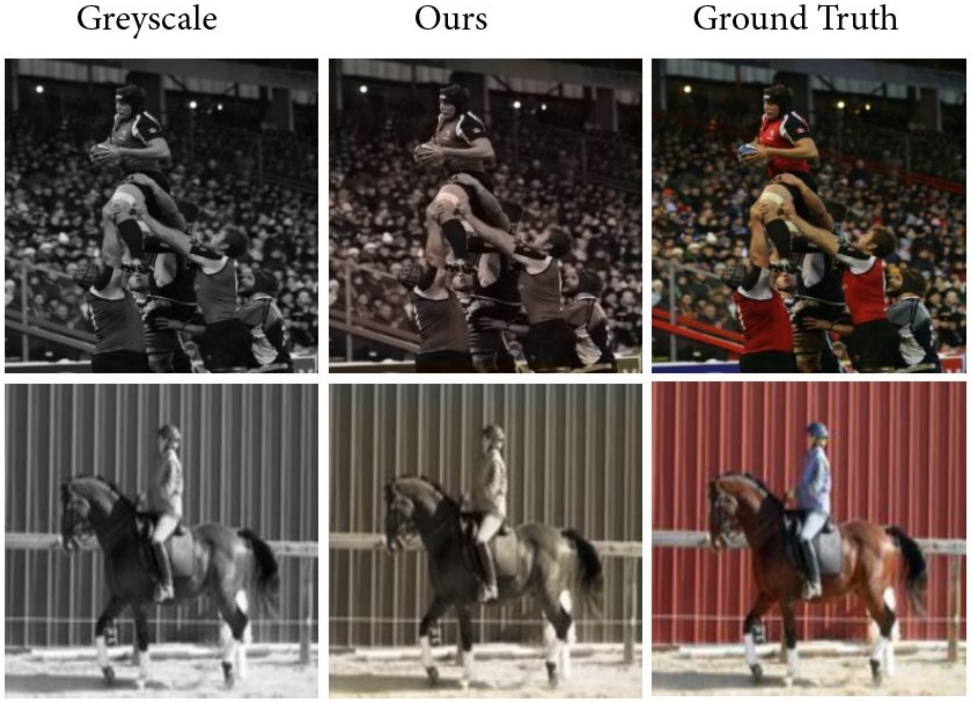
Experiments - 2000 Images



Performs good on images that have ‘dominant’ features (nature elements: grass and sky).

Not able to correctly color specific feature or object (ex. clothes)

Experiments - 10000 Images (TFRecords)



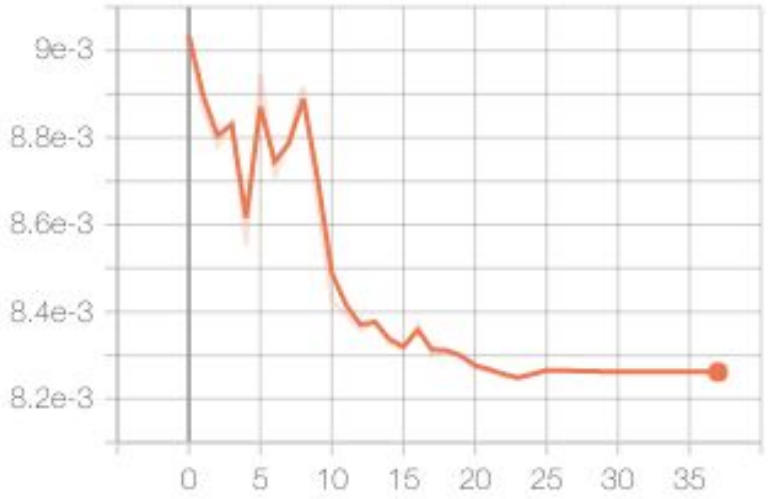
Hurdles encountered!

- Only able to correctly color large features or objects
- Brownish colorizations.
  - It’s the color that is most similar to all other colors, thus producing the smallest error.
  - Unsolved issue in papers as well
- Hitting plateau while training

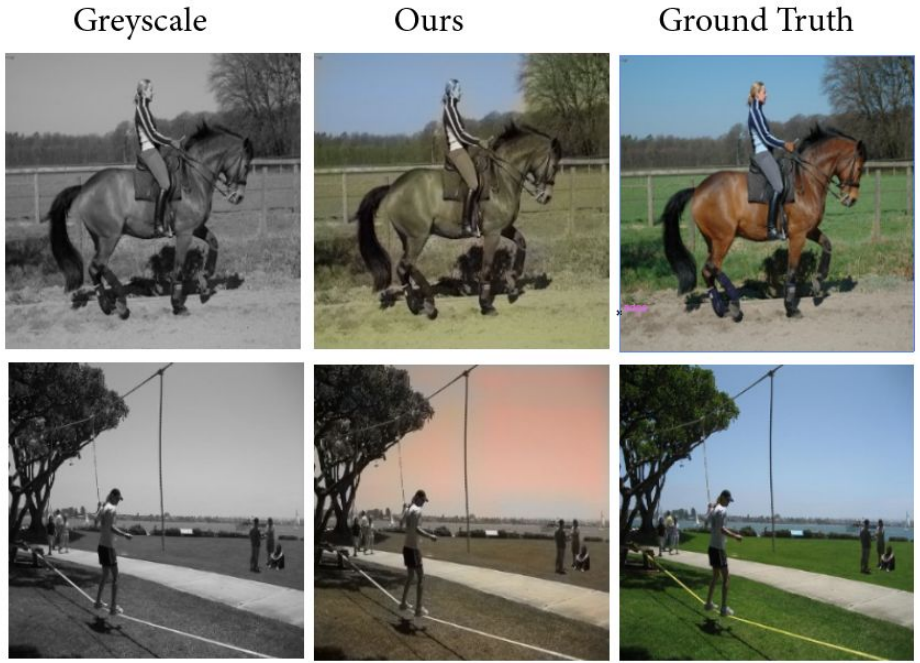
Loss

loss

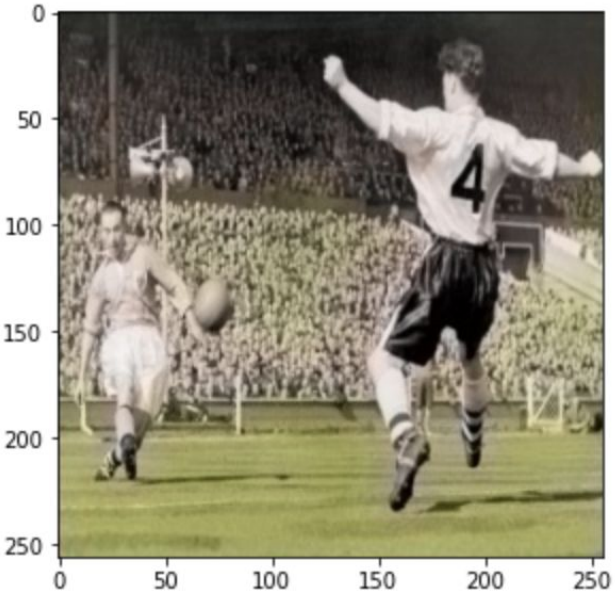
val_loss

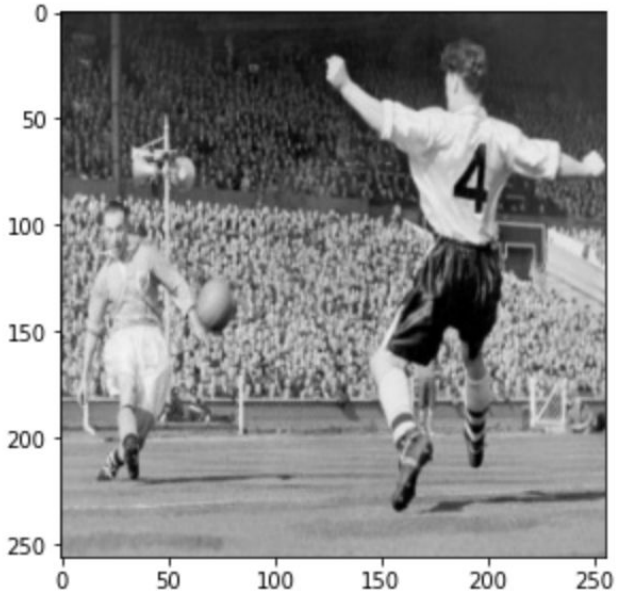


Miscolorization



Colorization on Old BW Picture





Conclusion

- Manage to achieve a quite good result in overfitted class
- Failed to get decent generalization
  - Small number of training data
- Precompute the embedding features and save the data into TFrecords to significantly reduce training time
  - There might be a bug in the implementation since the model could not learn properly and resulting in brownish color prediction

Learning Outcome

- Implement deep learning method
- know the workflow of creating deep learning model
- set up environment to develop deep learning model
- use pre-trained model to help the feature extraction process
- improve model performance
- Understanding GANS