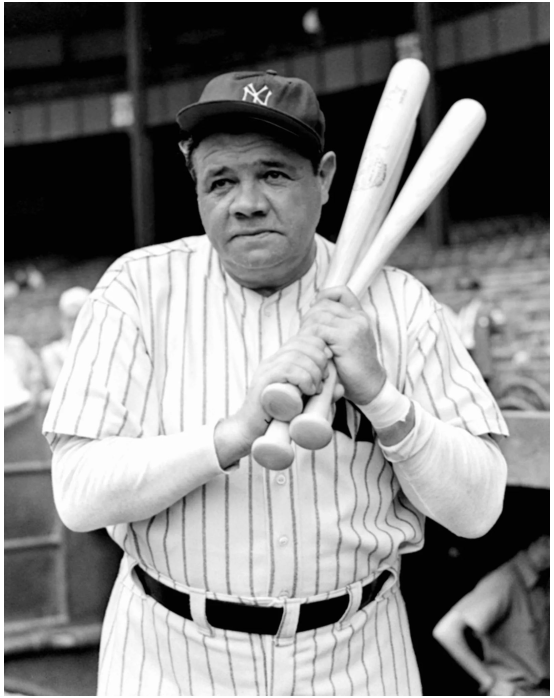
Image (Re)-Colorization Project

Donald Kane
Dakota Wilson
Jonathan Srinivasan

Project Idea

- Color imagery was not commonly used until the 1970s
- We want to find out how B&W sports images looked in real life
- Challenges
    - Quality and modifications of old cameras
    - Scope of photos fed into model needed to be adjusted in order to get accurate results. (ex: more landscape photos -> better landscapes)



Dataset

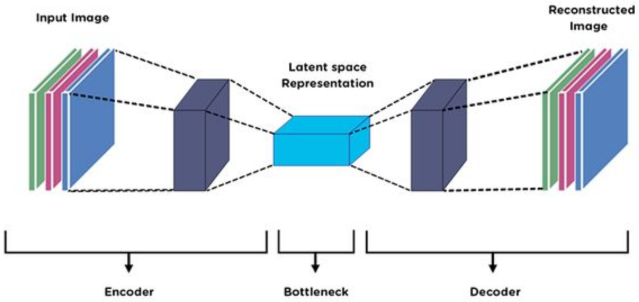
- Dataset of sports photos retrieved from Kaggle (408.4 MB)
- Sample of photos from 100 different sports, including some more obscure ones, 10,000+ entries

Formula 1 Racing

Curling

Basketball

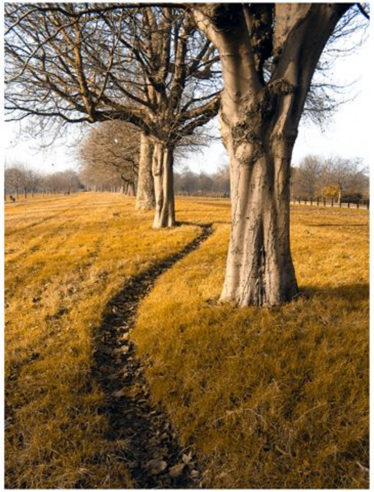
Methods



- Baseline: Convolutional Neural Network
- Decomposition of images into light portions
  - Lightness, green-magenta, blue-yellow
- Train models using b/w image as input feature, color portions as targets
- Our intention was to take this pre-existing model and improve upon it with more recent technologies

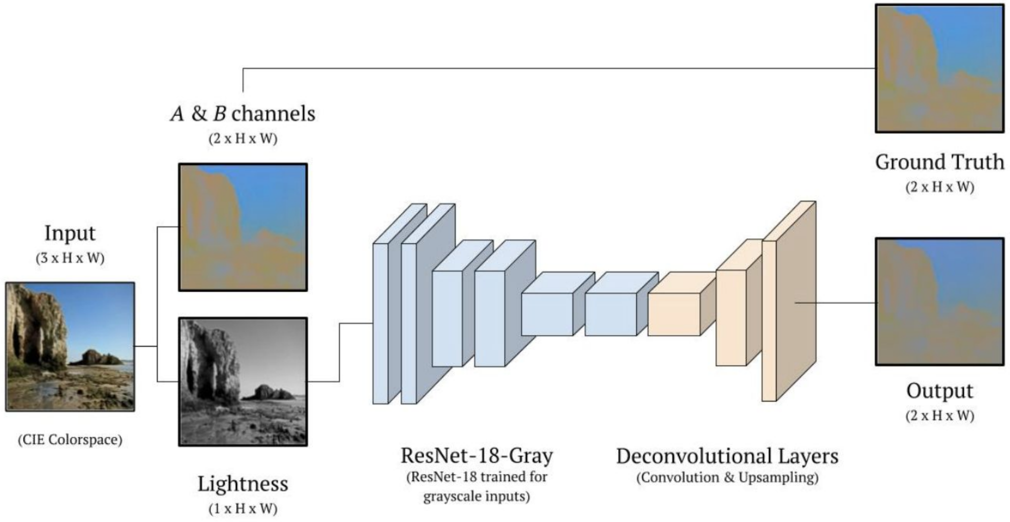
Pre-Project Expectations

- Pretrain models using our data
- Create python script that allows user to input any photo and have the pretrained models either:
    - Decompose a color photo and attempt to re-colorize it
    - Take in black and white photo and add color
- Potential avenue to explore: is it easier to recolor photos that are colorblind-friendly? (deuteranopia)



Baseline

- Used a convolutional neural net model from 2018
- 18 layers that reduce and convolute the photo, and then deconvolutes it back to a full size image
- Validates used MSE
- ReLU activation function used
- ADAM optimizer



Baseline Results

- Same validation image, different epoch results:
  - Epoch 8:

  - Epoch 90:

- Epoch 8 seems to be very little more than random guessing of colors, while epoch 90 (still not great) does manage to properly color some parts of the image

Model Training

- Original model was using Resnet-18, as well as some older packages for handling images
- Complete overhaul was done to the way images were loaded in using more recent and faster technology
- Loss function was changed from MSE - tended to choose ‘less wrong’ images that had little color
- Loss changed to Learned Perceptual Image Patch Similarity (LPIPS)
    - Places more of an emphasis on color being correct over valuing ‘intensity’ of color
- Minor changes made upon convolution layers in order to accommodate new image format as well as taking in non-batched data
- Due to change in loss function, it is difficult to numerically compare our results to our baseline, however, a human can easily tell the improvement

Model Output

Baseline vs Updated

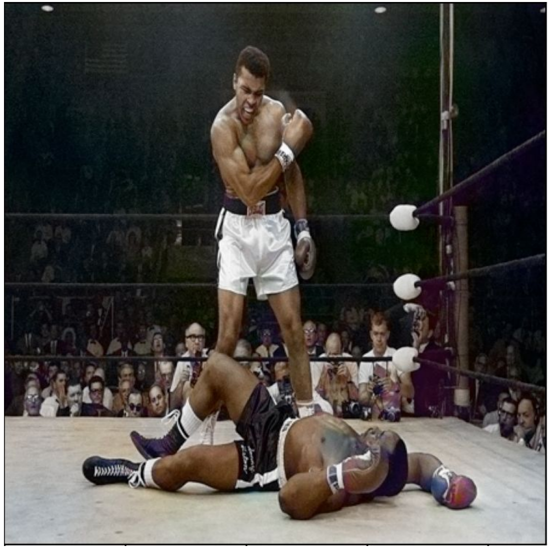
Original:	Baseline:	Updated:
		

Baseline vs Updated

Original:	Baseline:	Updated:
		

Demo

- Took the best weights and loaded into the model for use on our own photos



Challenges and Future

- DGX issues, such as accessing our folder with the saved model parameters
- Newer versions of Resnet are available, which have a lot more than 18 layers - potential for improvement but due to usage of only half of resnet-18 it would require a good bit of effort to update this
- Training time is lengthy - even with faster image loading technology
- New optimizer may yield improved results
- Different version of LPIPS loss may yield improved results

References

https://lukemelas.github.io/image-colorization.html

https://github.com/richzhang/colorization

https://cv-tricks.com/keras/understand-implement-resnets/

The Unreasonable Effectiveness of Deep Features as a Perceptual Metric
R. Zhang, P. Isola, A. A. Efros, E. Shechtman, O. Wang.

GitHub Repository

https://github.com/thedon58/DS5660_project