Polychromify

...

Deep Convolutional Autoencoder
for Landscape Image Colorization

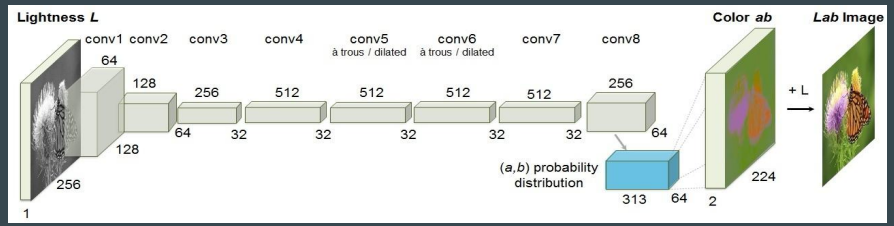
Image Colorization

- Hallucinating colors from black & white photos
- Useful to revive historical photos & films with colors
- Difficult task even for humans (ill-posed problem)
- Very challenging & fascinating problem

Learning-based colorization approach

ECCV16

- Zhang et al. 2016
- Convolutional Neural Network
- Classification setup + class rebalancing
- Rich colorization results

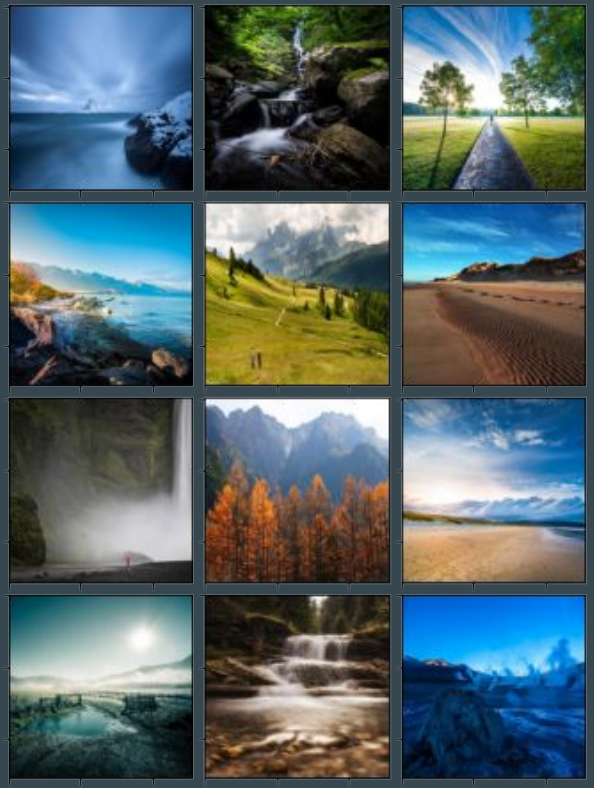


SIGGRAPH17

- Zhang et al. 2017
- CNN + *user colors hints*
- Merge automatic + user input approach
- Several plausible colorization + more realistic results

Dataset

Landscapes Dataset - Sample Overview

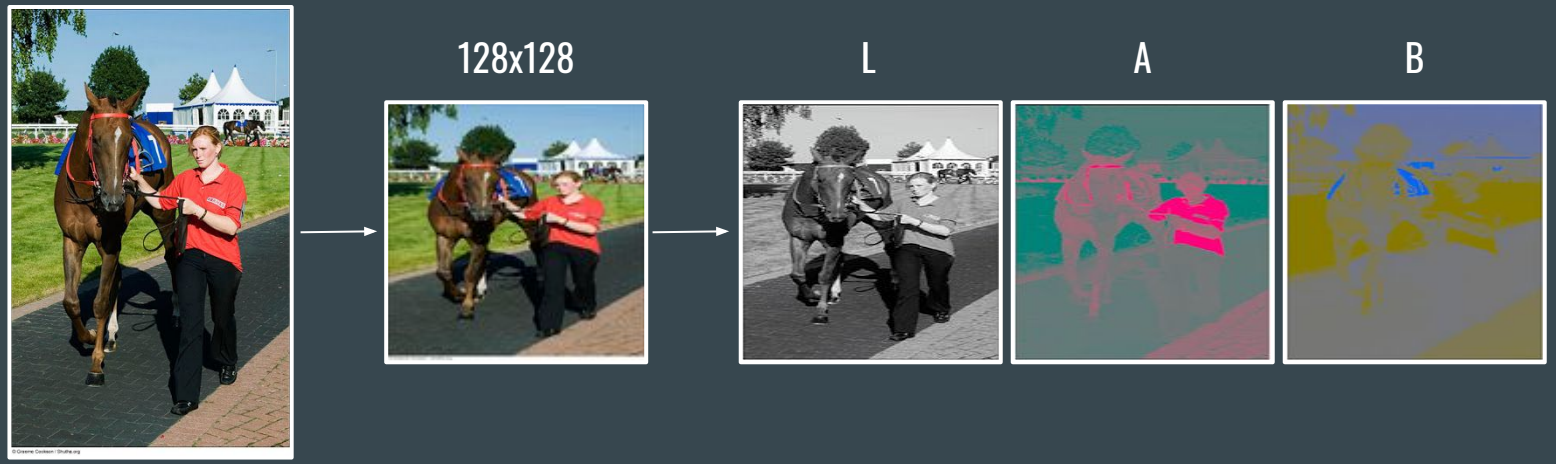


Features:

- 4.3k total images
- Diverse landscapes categories
- Different sizes with high resolution
- Dataset split
  - 2752 images (64%)
  - 688 images (16%)
  - 860 images (20%)

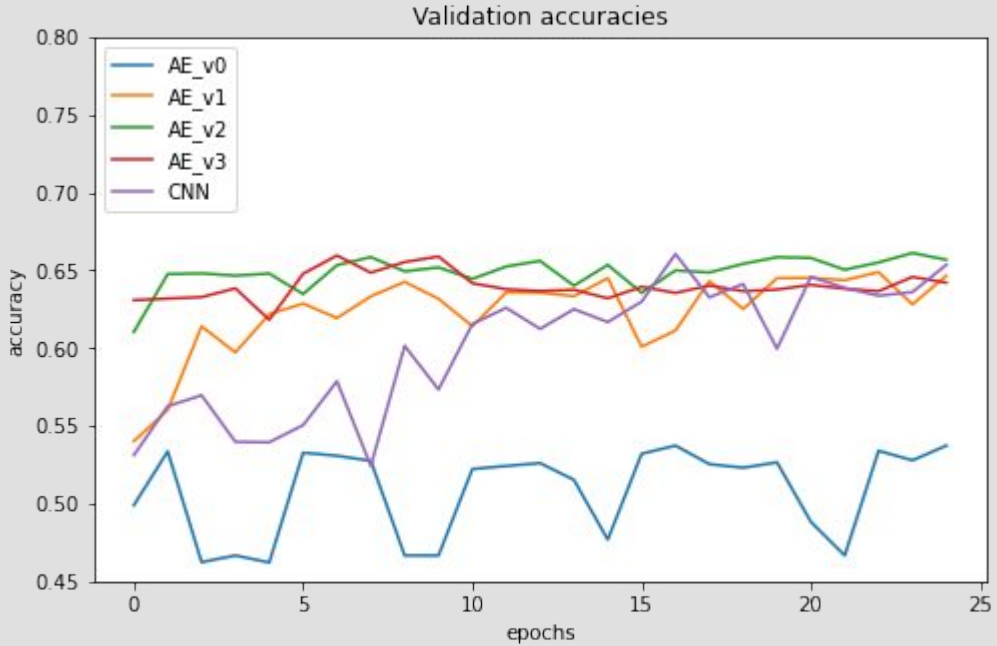
Pre-processing

1. Resize to fixed size of 128x128 pixels
2. RBG to CIELAB color space conversion (closer to human perception + easier to separate)
3. Normalization of AB channels to range [-1, 1] for better training (avoiding saturation activation function)



Model selection

1. Vanilla Autoencoder (v0)
2. Convolutional Autoencoder (v1)
3. Convolutional Dense Autoencoder (v2)
4. Deep Convolutional Autoencoder (v3)
5. CNN



Convolution is key!

Best Architecture :

Convolutional Dense (v2)

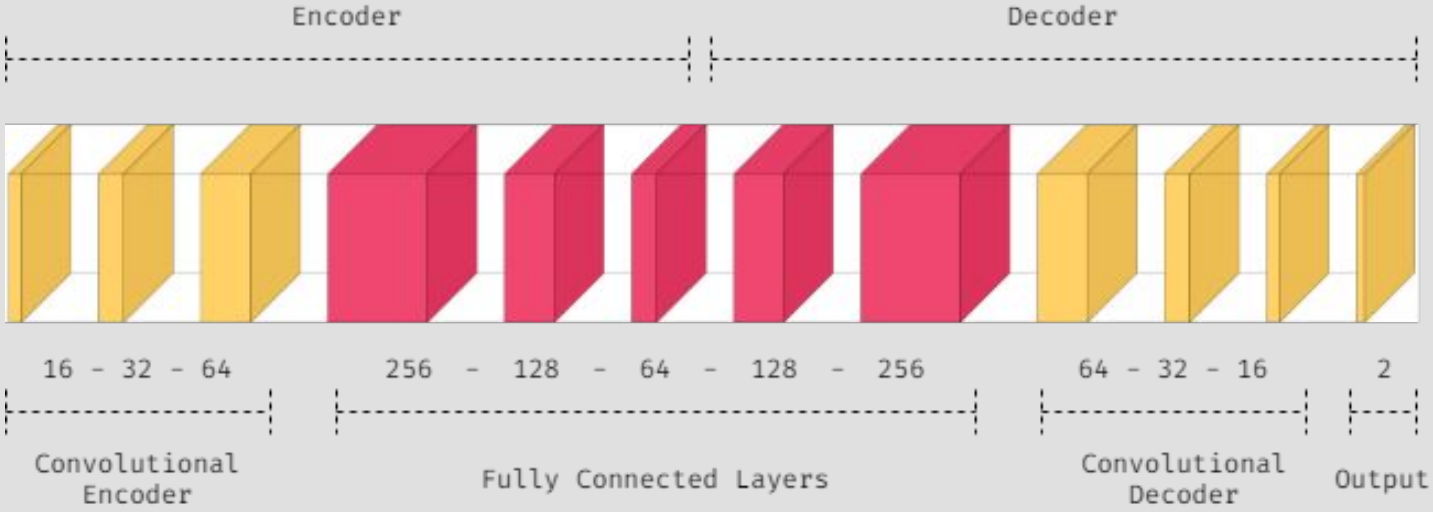
Hyper-parameter tuning

Hyper-parameters						
Models	Neurons*	Filters**	Strides/Upsampling	BN	Accuracy	Time
v1	Low	Low	No	No	0.6814	35 min
v2	Medium	Low	No	No	0.6792	52 min
v3	High	Low	No	No	0.6851	1h 5 min
v4	High	High	Yes	No	0.6727	1h 45 min
v5	High	High	Yes	Yes	0.6527	1h 52 min

** Neurons : Low (64 - 32 - 16 - 32 - 64) | Medium → (128 - 64 - 32 - 64 - 128) | High → (256 - 128 - 64 - 128 - 256)*
*** Filters: Low (16 - 32 - 64 - 64 - 32 - 16) | High → (32 - 64 - 128 - 128 - 64 - 32)*

Polychromify Architecture

Deep Convolutional Dense Autoencoder



Quantitive results: accuracy

CNN

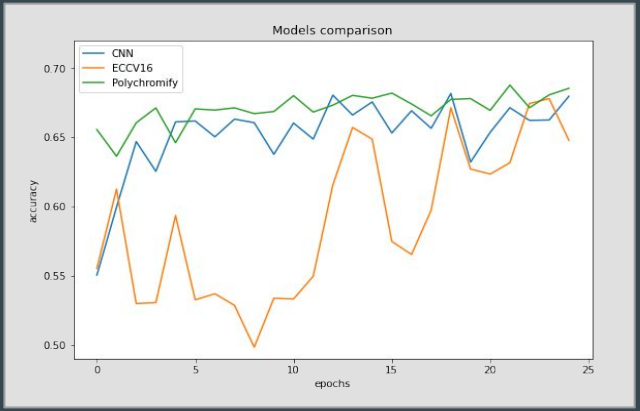
- Baseline model
- Tradeoff complexity vs accuracy

ECCV16 (trained from scratch)

- Powerful but high number of parameters
- Slow to train
- Data hungry model

Polychromify

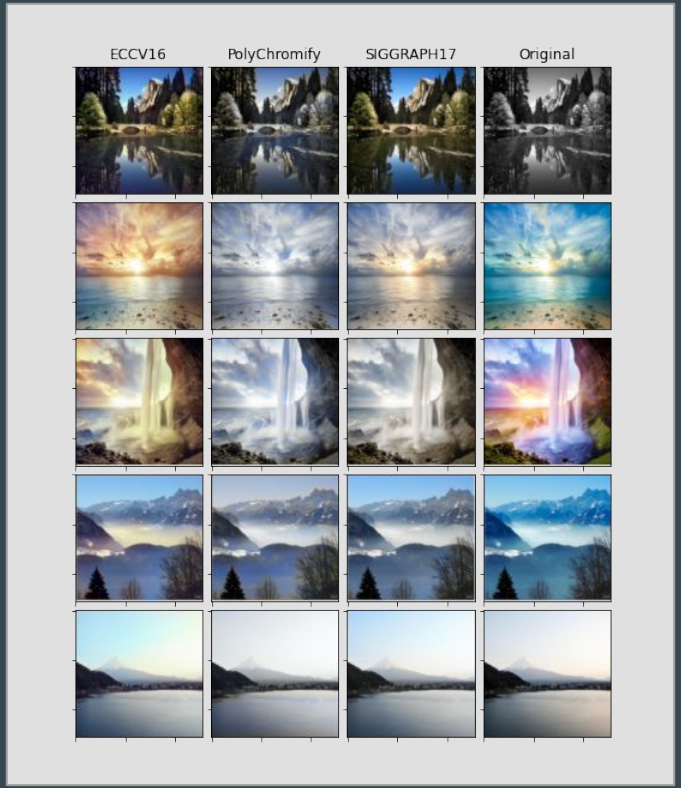
- Best overall
- Fast training
- Tuned on this specific dataset



Final results on test set			
Model	Parameters	Training Time	Accuracy
CNN	98×10^3	2h 10 min	0.6535
ECCV16	32×10^6	6h 40 min	0.6400
Polychromify	293×10^3	1h 40 min	0.6845

Qualitative results: colorization

- ECCV16 (pre-trained)
  - Vibrant colors
  - Over-saturated predictions
- SIGGRAPH17 (pre-trained)
  - Less saturated predictions
  - More realistic colorization
- Polychromify
  - “Shy” color predictions,
    (most from yellow to blue range)
  - Difficult to generalize from bluish pictures



Final Remarks

Contributions

1. Simple Convolutional Autoencoder architecture
2. Fast training + low number of parameters

Future Improvements

- Improve dataset by rebalancing landscape categories
- Train on larger dataset (e.g. ImageNet) for better generalization
- Pre-trained models for object recognition

The End

...

Github Repository
https://github.com/davide97g/polychromify