



Color black and white images and videos automatically!

## Introduction:

Mughal-e-Azam, one of the greatest films ever made in India, was shot in black and white. Years later, on popular demand, it became the first Hindi film to be digitally colorized. The method they followed to color it was manually coloring each and every pixel of each and every frame. As can be imagined this is a tedious, time and money consuming process. Wouldn't it be wonderful if we let the computers color it and all we had to do was just a mouse click? VanGogh does exactly the same-- with just one click it can color a photograph or a video.

VanGogh is designed to colour grayscale pictures using deep learning algorithms. The system is implemented as a feed-forward pass in a CNN at test time and is trained on over a million color images. VanGogh is primarily implemented in python using caffe as the main deep learning framework and Tkinter as GUI.

The goal of VanGogh is not to recover original colours from a grayscale image but rather to produce a plausible colorization that is close to real world representation of the object in focus. vanGogh is extended to not only color pictures but also black and white videos using openCV python library.

Applications of such software can be many. As mentioned above it can be used to color entire movies. It can also help us in enhancement CCTV footage, microscopic and macroscopic photography.

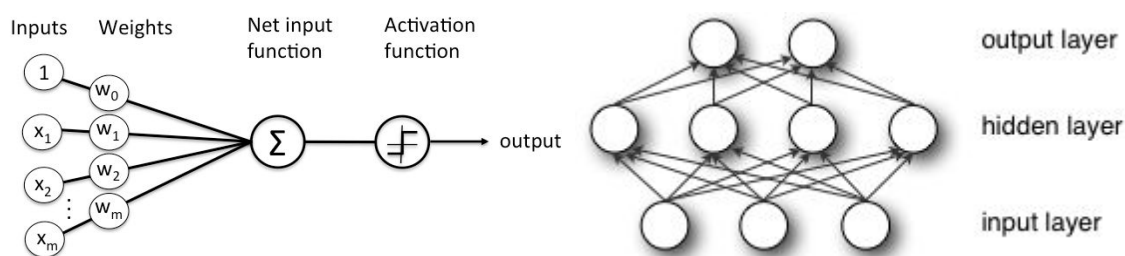
## Description:

There are various colorization strategies that can be adopted to color a black and white image. One of them (which is implemented in VanGogh) could be assigning color to the

image - pixel by pixel, analysing the intensity. The color is given to the image by learning from millions of other color images with similar content (training dataset).

In this way we create a "Deep Neural Network" with a lot of digital neurons where neurons react differently to different combination of inputs. Each neuron in this network contributes to the prediction of total amount of a certain color in the picture.

We do not define any output of these neurons but rather we let the system learn by itself during the "training process". Such a network consists of millions of neurons with several neural networks where the output of a layer is the subsequent layer's input. Each layer is made of nodes. A node is a place where the computation happens. It takes in a set of inputs along with coefficients as weights and gives an output accordingly.



Images source: Skymin<sup>1</sup> (Left: Node, Right: Network layers)

The neural network used in VanGogh is trained using several thousands of photos. VanGogh is implemented in Python 2.7, using numpy, scikit-learn, OpenCV libraries and caffe.

## How VanGogh works:

VanGogh has a graphical user interface built using Tkinter in Python. The user can browse and select the picture he/she wants to color. The selected picture is then colored using caffemodel trained by researchers at Berkeley.

---

<sup>1</sup> <https://deeplearning4j.org/neuralnet-overview>

### Steps to use VanGogh:

1. Required packages: numpy, scipy, caffe, PLT, Tkinter, opencv, matplotlib.pyplot, skimage.color, ntpath, Tkconstants, tkFileDialog

2. Install the required packages on the PC.

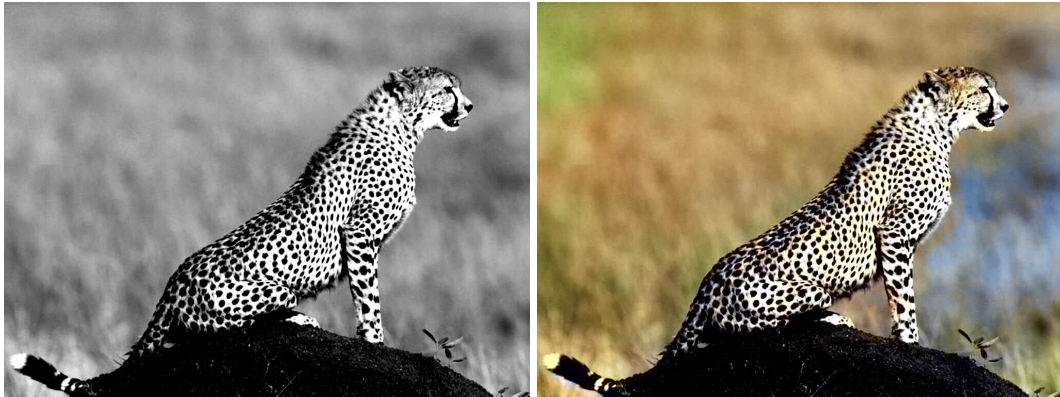
3. To color an image:

- Execute "gui.py": opens an application named "VANGOGH"
- Click browse and select the black and white image
- The image is colored and displayed in the application along with the black and white image
- To see the black and white image click "SHOW B/W" and to see the colored image click "SHOW COLOR"

4. To color a video:

- Create the frames: execute create\_frames.py. This takes video file (name.mp4) as argument.
  - (Eg: python create\_frames.py video.mp4)
  - This breaks the video into frames and saves them in the folder named "frames"
- Color the frames: execute color\_frames.py. This takes number of frames as argument.
  - Check the number of frames in the folder "frames".
  - (Eg: python color\_frames.py 100)
  - This colors the frames that are in the folder "frames" and saves them in "colored"
- Get the video (colored): execute create\_video.py. This takes number of frames as argument.
  - (Eg: python create\_video.py 100)
  - This creates the video from the frames in the folder "colored" and saves it in "colored\_videos"

## Results:



We have shown that colorization with a deep CNN can come closer to producing results indistinguishable from real colour pictures and videos. Although only trained to colour, the model network learns object representations that can be used for object identification, segmentation and detection.

## Resources:

Caffe:

<http://caffe.berkeleyvision.org/>

<https://github.com/BVLC/caffe>

For coloring:

[https://github.com/richzhang/colorization/blob/master/demo/colorization\\_demo\\_v2.ipynb](https://github.com/richzhang/colorization/blob/master/demo/colorization_demo_v2.ipynb)

Model used:

[http://eecs.berkeley.edu/~rich.zhang/projects/2016\\_colorization/files/demo\\_v2/colorization\\_release\\_v2.caffemodel](http://eecs.berkeley.edu/~rich.zhang/projects/2016_colorization/files/demo_v2/colorization_release_v2.caffemodel)

Other links:

<https://github.com/richzhang/colorization>

<https://deeplearning4j.org/neuralnet-overview>

## Acknowledgements:

Dr. Yamuna Prasad, Dr. Ravi Kothari and Dr. Raghavendra Singh

## Team Members:

Deva Surya Vivek, Chaitanyashareef Kulkarni and Paras Bhattra