

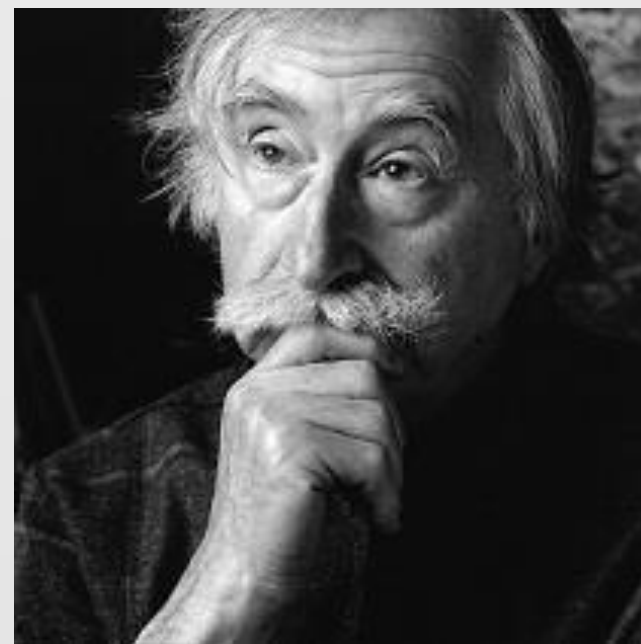
# Digital Analysis of Paintings using TensorFlow

By Lasse Falch Sortland (Laf15)

## Aim of the project

The aim of this project is to develop a Convolutional Neural Network (CNN). It will be able to take an image of a painting and classify as either being of a specific painter or not. Currently I am using a dataset of Sir Kyffin Williams (KW) paintings, but more may be added as time goes by. The process the images go through is 3 steps.

Figure 1 [1]: **Sir Kyffin Williams**. This is the artist behind the art that my convolutional neural network is currently being trained on.



As time goes by the aim is to add a Neural Style Transfer (NST) to get the ability to input images that are not of the KW style and add that style to it. NST uses the information that the neural networks can take out of a Neural Network (NN), usually a CNN, and transfers that to another image.

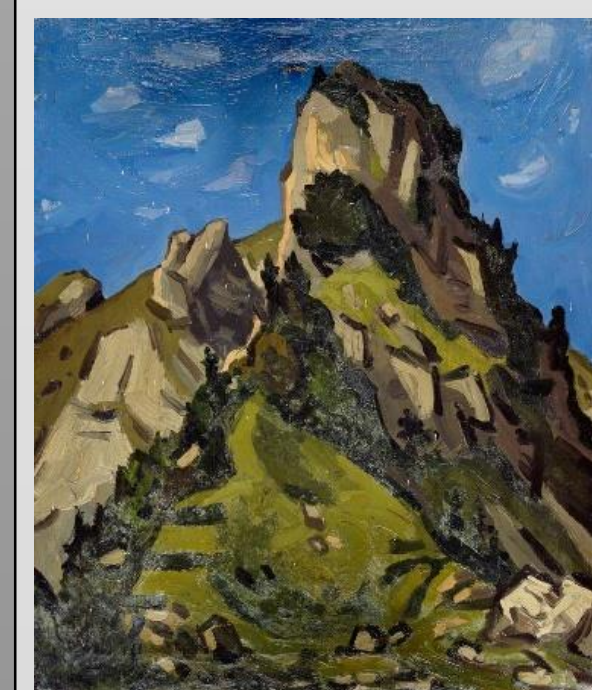


Figure 2 [2] [3]: **Alpine Mountain Landscape by Kyffin Williams**. This is one of the images in my dataset. As you can see the style of the image is quite idiosyncratic as he uses a palette knife rather than a brush.

Most of the background work has been reading up on how to use TensorFlow(TF) and trying to figure out a way to make it work using the MNIST dataset. One of the biggest problems with this project so far has been the documentation of TF. There are a severe lack in proper, well structured documentation which makes it extremely hard to figure out how to fix something when you do something wrong. On top of that the error messages are very often quite ambiguous and makes for even harder to get a working piece of software.

## Progress

The current working version is used to map integers from handwritten digits in the MNIST dataset. I am currently working on spoofing the dataset object that TF uses for the MNSIT dataset, as most tutorials online use the MNIST dataset object which is a different dataset object than the one you get if you read in the files how you are supposed to do.

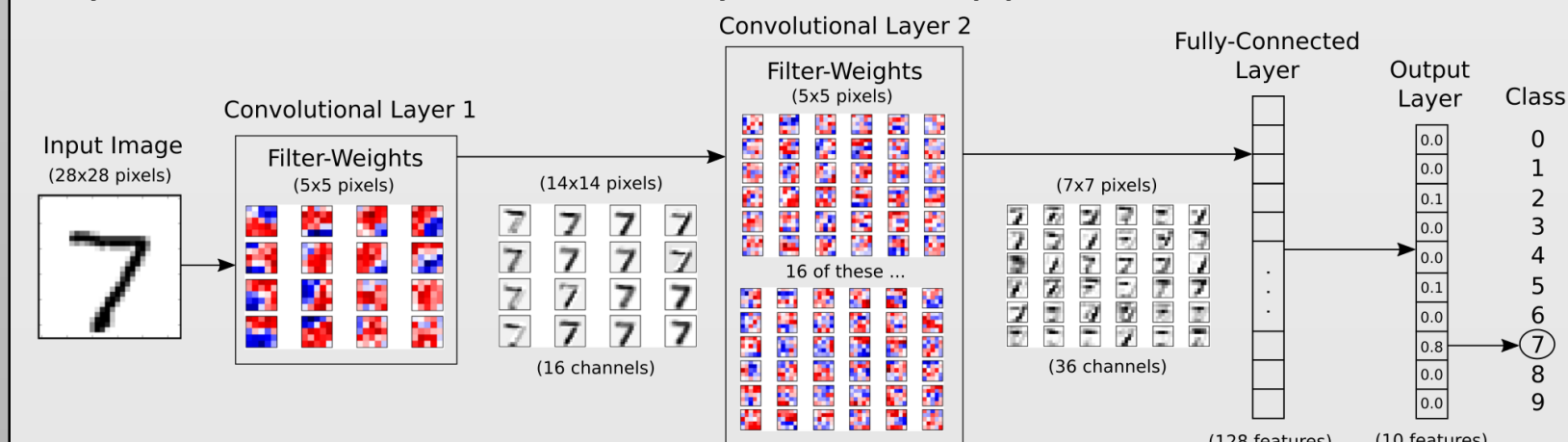


Figure 3 [4]: MNIST flowchart by Hvas-Labs. This image is the process of what goes on in the CNN that Hvas-Labs Designed for Hand Writing Recognition. The input layer is an image of what you wish to detect, in this instance it's a hand written 7. The Convolutional Layer 1 is the filters that the Convolutional Network get out of that image after it's changed it's weights based on said image. The 14x14pxl image set is a representation of what those filters see. The Convolutional Layer 2 is the second layer of convolutions and represents the filters. The Fully Connected Layer has 128 neurons and that feeds into another fully connected layer called the output layer. The output layer has the certainty of a image being of a specific class in a array with as many classes as you have. For this it might be

0, 0.008, 0.171, 0.008, 0.003, 0.009, 0.017, 0.038, 0.728, 0.015, 0.003

So you can see there is some confusion between 1s and 7s however the CNN is 72.8% sure that it's a 7 in this instance.

## Further information

[1] Krizhevsky, Sutskever and Hinton [08/12/12]. ImageNet Classification with Deep Convolutional Neural Networks [Online]. Downloaded: [21/02/18]. Available: <https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>  
[2] Anish Athalye [12/10/12]. neural-style [Online]. Downloaded: [08/02/18]. Available: <https://github.com/anishathalye/neural-style>  
[3] Google LLC, Vincent Vanhoucke, Arpan Chakraborty [N/A]. Deep Learning [Online]. Downloaded: [29/01/08]. Available: <https://eu.udacity.com/course/deep-learning--ud730>  
[4] No one author [N/A]. Kyffin Williams [Online]. Downloaded: [08/03/18]. Available: [https://en.wikipedia.org/wiki/Kyffin\\_Williams](https://en.wikipedia.org/wiki/Kyffin_Williams)  
[5] A. D. Brown, G. L. Roderick , H. M. Dee and L. Hughes [2016] Visual digital humanities: National Library of Wales [Online]. Downloaded: [08/03/18]. Available: [https://books.google.co.uk/books?hl=en&lr=&id=3AcUDgAAQBAJ&oi=fnd&pg=PA89&dq=%22Visual+digital+humanities%22&ots=BkBUhFhpg\\_&sig=Dx1661MO5PU24G3VIVhWB0fv5rM#v=onepage&q=%22Visual%20digital%20humanities%22&f=alse](https://books.google.co.uk/books?hl=en&lr=&id=3AcUDgAAQBAJ&oi=fnd&pg=PA89&dq=%22Visual+digital+humanities%22&ots=BkBUhFhpg_&sig=Dx1661MO5PU24G3VIVhWB0fv5rM#v=onepage&q=%22Visual%20digital%20humanities%22&f=alse)

## Technical Information

The project is using the TF libraries as the library for doing NN related programming however it also uses OpenCV for image processing, sklearn for some machine learning functions, Matplotlib for plotting and Numpy for large, multi dimensional arrays. The bulk of the project will be written in Python 3.5 using TF version 1.5.0.

The object I have added has required me to figure out how the tensorflow.contrib.learn.python.learn.datasets.mnist.dataset object works and spoofing what it does to add my dataset to the CNN rather than the MNIST. The way the objects are supposed to be read into Tf is using a TFRecordReader on .tfrecord files which have a complicated structure that have next to no official documentation on how to use, how to create or how to understand. I have been in contact with someone else who does a similar project to mine, who have also been having the same problem with the .tfrecord files and I have decided to scrap tfrecords and spoof them instead.

So far in this project I have spent about 30% of my time reading up on how to do something in tensorflow and another 30% of my time has been spent on trying to actually make that work realizing that one of the things has been changed or deprecated making the whole project wasted as there is no mention of what python version is used nor what TensorFlow version is used.

For more information about processes and the latest look at what this project is all about visit

<http://kallah.co/>

## Work

The first thing I need to do is to make my dataset object work, currently it's accepting the images but not the labels but I have had some problems with understanding the errors I've been given. When the CNN accepts my dataset I need to start working on optimizing it. When I have finished that part I will start working on getting the NST up and going to get some style back out of the system.

## Future Work

For future work when I finish I might try to do some image processing prior to feeding it to the network to maybe try to highlight the style more or something more abstract like just see what happens if I take the two images and get the average of them or blur them or maybe transform them to look different and see how that would affect the output both the CNN and the NST



Figure 4[2] [3]: Examples of image processing methods I might use

From top right being 1, to top left being 2 middle right being 3:

2 ) Original with gaussian blur  
3 ) Original sharpened and darkened  
4) Original sharpened and darkened and then

divided by original

5 ) Original with gaussian blur subtracted by original

6 ) Original sharpened and darkened divided by original with gaussian blurred subtracted from original

Here you can see some of the things you can do to images to get features out of them, four, five and six all are different edge maps and would make for a quite interesting style to add to an image if you imagine that the Black in five and six is treated as alpha and the white in four is treated as alpha.