The Art Thief – Is this a Whiteley?

An artist classifier using Convolutional Neural Networks (CNN)

Context and Background.

Brett Whiteley AO (1939-1992) [1] was one of Australia's most influential and famous artists. As well as his contribution to Australian and international art, his life and works were surrounded by intrigue.

In 2015, two art dealers were charged with forging Whiteley's works and selling them for millions [2].

The art dealers were consequently acquitted [3]. This project investigates the possibility of using machine learning to classify artwork images as belonging to the artist in question.

This project is of interest to artists and art lovers, as well as people interested in image classification with small datasets.

Figure 1 - "The Balcony 2" Brett Whiteley (1975) AGNSW (top).

The alleged fake Brett Whiteley fake painting (bottom)

Domain Background.

This problem is in the domain of machine learning, specifically classification and image processing. Some previous works include identification of graffiti artists, [3], image classification using CNNs [4], [5], [6], AI artwork synthesis [7] and formation of new artworks through combinatorial synthesis [8].

Problem Statement.

Going from high level into detail, the problem to be solved is:

“Is this an artwork by Brett Whiteley?”

The problem can be broken down into a machine learning problem using a system engineering approach.

Due to the sparse nature of the alleged fakes (2-3) training a classifier on these fake images was unlikely to prove fruitful.

The problem then moved to the generation of a binary classifier (Whiteley/non-Whiteley) for which the fake painting was evaluated. Identifying Whiteley's "style" in the classifier is proposed to assist with the classification of the true paintings.

There were several constraints relating to this problem, detailed below.

<u>Data Sparsity</u>

While there are hundreds of thousands of paintings available, there are only several hundred works completed by Whiteley in his lifetime. This results in sparsity of data.

<u>Unbalanced Datasets</u>

an unbalanced dataset (too many non-Whiteley paintings relative to Whiteley’s) results in issues including overfitting and stuck optimisation [9]. Balancing the dataset by reducing number of non-Whiteleys now places an upper bound on the accuracy that can be achieved. Approaches to resolve this unbalanced issue are described further in this report.

<u>Ground Truth (lighting, pose, scale)</u>

Ideally there would be good ground truth data available (for instance the paintings were photographed with the same scale, location, lighting condition, pose, resolution). However, the images obtained for the classification dataset were photos of the paintings from the web with varied scale, pose, resolution, and lighting.

<u>Image shape and size</u>

All of the images have different formats, shapes, sizes, and squareness. Some processing will be required to ensure uniform piping into the ML model.

<u>Artistic Media</u>

Whitely used various media (ink, colored paint, sketch) and elements in one media may have high saliency with elements in the same media painted by a different artist.

Some clues to his unique artistic style include the use of the color blue [10].

<u>Veracity/Provenance</u>

It is taken on face value that the photographs of the art works are indeed true copies of original works by the artists, including Whiteley. If this assumption proved invalid it could render the ML model invalid.

Datasets and Inputs

Prior to the proposal, a dataset of images was prepared as there was none available for the purpose of this classification example. Creation of the appropriate dataset is a key part to solving this problem and also has a large potential for flawed or inaccurate results.

By collating artists of similar styles in the non-Whiteley dataset, it was hoped that additional discrimination of "style" would be achieved relative to a normalised artistic dataset (for constant balance).

Dataset:

Labelled (1): Brett Whiteley paintings and sketches.

Labelled (0): other artworks and sketches from other artists with similar style. Photographs of the subjects painted by Whiteley. Rationale for the selection of the artists from this set is provided below.

Artists in the (0) dataset:

Henri Matisse – Whiteley drew a lot of influence and style from Matisse's sketches and paper cutouts.

Pablo Picasso (sketches only) – this is to assist the ML model in separating out stylistic lines that produce a different result between the artists.

Lloyd Rees – Whitely admitted being influenced by Rees [11] and there are many of Rees' artworks available through the AGNSW website.

Mark Rothko – this is to assist the ML model differentiate in the situations where there are large blocks of colour for which both Rothko and Whiteley were famed.

Francis Bacon – Bacon was a contemporary of Whiteley and their works are often compared.

Vincent Van Gogh – Whiteley was influenced by Van Gogh's work and they both painted the same subject "The night café". Both are included in the dataset.

Harry Kent and Kristin Hardimann – both are successor artists of Whiteley and his styles and subject influenced both.

During the training of the CNN, it became clear that the original dataset was imbalanced and the model became "stuck" at a fixed loss. The (0) dataset had elements removed until a balance was achieved. Details are discussed further in the report.

Solution Statement

The images are pre-processed to fixed size and converted into arrays or lists for processing.

The image data structures are investigated following the artist's style cues - use of block blue color and solid paintings, along with sweeping strokes on sketches.

The data is assessed for salient features (blue color, image histogram peak height) and the Receiver Operating Characteristic (ROC) and Area Under the Curve (AUC) [16] metrics are employed to understand if a simple classifier based on salient features is adequate.

Further, the data is investigated to see if relevant feature extraction or "style identification" can be performed on the dataset. This is performed via feature mapping using the ORB (Oriented Fast and Rotated Brief) algorithm.

This investigation proved informative but inadequate, so unsupervised learning in the form of Convolutional Neural Networks were employed.

The first solution was to use a convolutional neural network (CNN) to effectively classify the artwork as a Whiteley / non-Whiteley. This CNN was trained from scratch on the dataset. Techniques were employed to resolve the unbalanced and sparse nature of the dataset.

The second solution investigated was to use transfer learning using a pre-trained model on the ImageNet architecture (eg. VGG16 [12]). One issue with this approach is that the model had been trained on large classes of objects (cats, dogs, cars) but not on artists styles or paintings. Recent work on artistic style transfer using pre-trained models such as VGG16 and Xception [13] show that this is not a fruitless endeavour.

Cross-entropy, MSE and log-likelihood error function were evaluated in the classifier.

The dataset was broken into separate training, validation and test sets, along with a final test which is the proposed forgery for evaluation.

Implementation

Implementation in python + keras + Tensorflow as a Jupyter notebook. Image processing was performed using python cv2.

The code was executed in an Anaconda environment, on a local machine. A GPU and AWS instance was instantiated as part of the investigation.

There was a significant amount of work to create an image pipeline that converts all of the images into appropriately sized, oriented, correct aspect ratio files along with an appropriate data structure containing the associated meta-data / labels for processing.

Benchmark Model

There are multiclass artist classifiers using different techniques that exist in the literature.

The goal is to achieve similar to state-of-the-art binary classification which is the order of 85% or more for such a small dataset [14], [15].

Evaluation Metrics

The null hypothesis answers the question “what is the likelihood that this is NOT a Brett Whiteley artwork?”

The evaluation metric for the salient features is the ROC-AUC score [16]. A value above 0.5 shows that the machine is able to perform a classification better than random.

The evaluation metric for the CNN (scratch-built and transfer) is classification accuracy.

Data Exploration

Human Interaction with the data

Many of the painting images were viewed qualitatively. This was to get a feeling for the problem before diving into the data analysis. Whiteley's paintings appeared to be characterised somewhat by large blocks of color, small image sections as well as sweeping lines. It was rare to see orthogonal lines oriented with the pose of the viewer or high frequency detail. This gave some clues for the analysis and data exploration.

Image Histogram peak

An image and its histogram are shown below. A hypothesis was formed that the images in the two classes of the dataset might have sufficiently different image intensity histogram peak distributions to perform a classification.

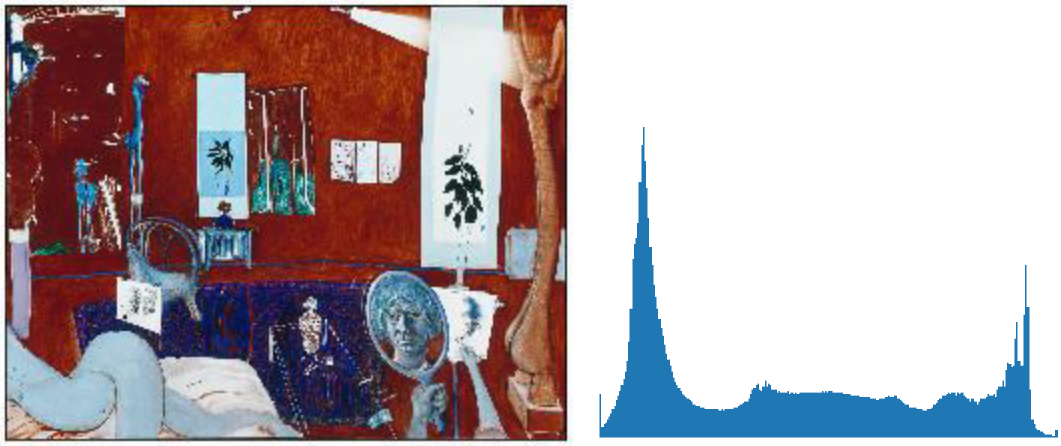


Figure 2 - Whiteley painting image and its histogram

The images from both classes were converted to grayscale images of identical size and resolution. From this the normalised image intensity histogram, the peak location for all images in the two classes of the dataset were obtained (two lists). This distribution histogram was created for the two classes and provided in the image below.

From the data it could be seen that the paintings in the Whiteley class typically had a higher peak intensity histogram value relative to the non-Whiteley class, suitable for some classification.

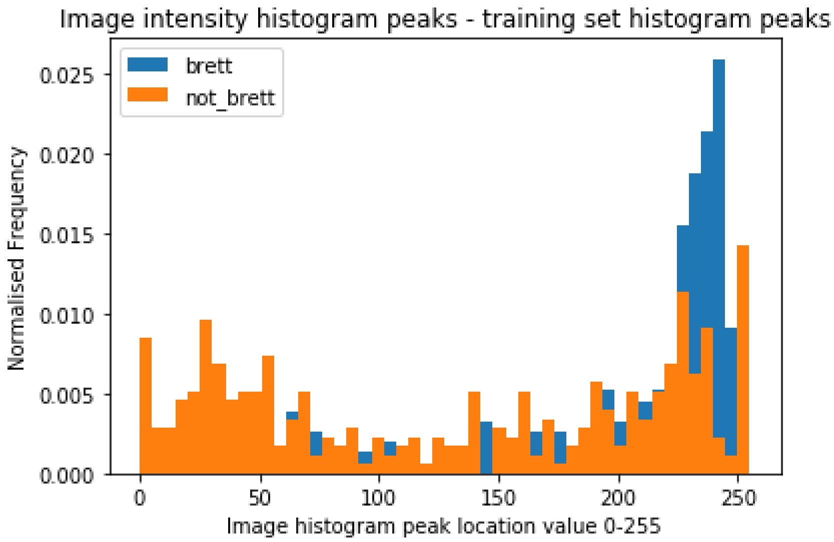


Figure 3- Image Intensity histogram peak distribution per class

To evaluate the performance of the model, the ROC-AUC characteristic was developed as a function of the classification threshold (from 0 being all non-Whiteley to 254 being all Whiteley).

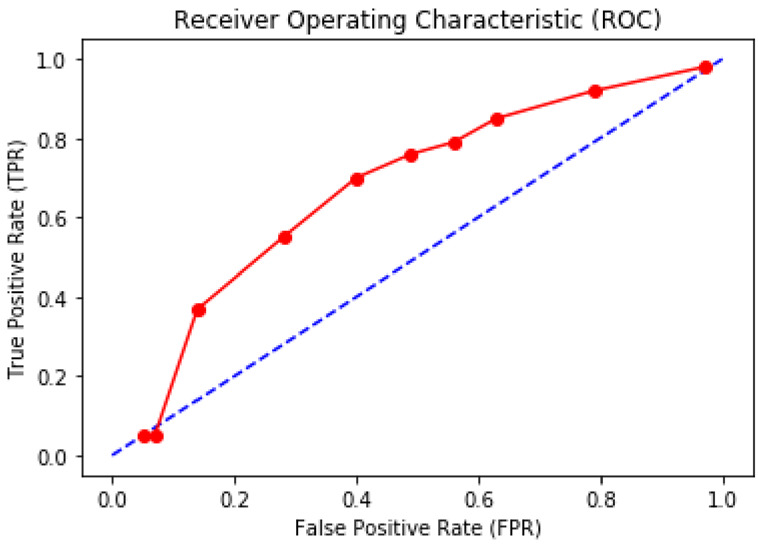


Figure 4 - ROC-AUC for the intensity histogram

The performance result was an AUC of 0.645 which is better than random selection. It might be suitable for pre-selection of images but was not at a sufficiently accurate level to credibly detect a fake. Nevertheless, this level of performance for a classifier would be comparable to many people's ability at an art gallery.

<u>Color histogram</u>

Whiteley's use of blue in his artworks gave a clue toward another classification model. A further hypothesis was formed that the images in the two classes of the dataset might have sufficiently different color histogram peak distributions to perform a classification.

The images from both classes were converted to images of identical size and resolution. The RGB elements from the image was further extracted and the Blue co-ordinate further extracted.

From this the Blue color co-ordinate histogram, the peak location for all images in the two classes of the dataset were obtained (two lists). This distribution histogram was created for the two classes and provided in the image below.

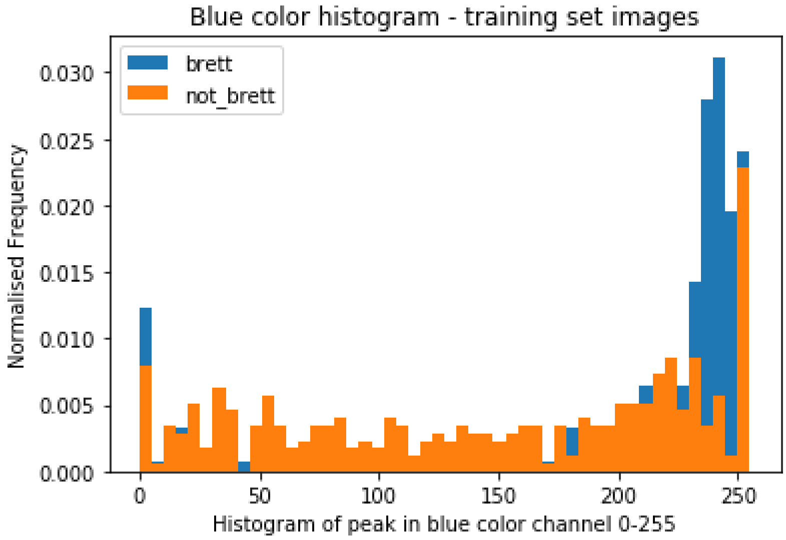


Figure 5 - Blue color intensity distribution per class

From the data it could be seen that the paintings in the Whiteley class typically had a Blue content value relative to the non-Whiteley class, sufficient for some classification.

To evaluate the performance of the model, the ROC-AUC characteristic was again developed as a function of the classification threshold (from 0 being all non-Whiteley to 254 being all Whiteley).

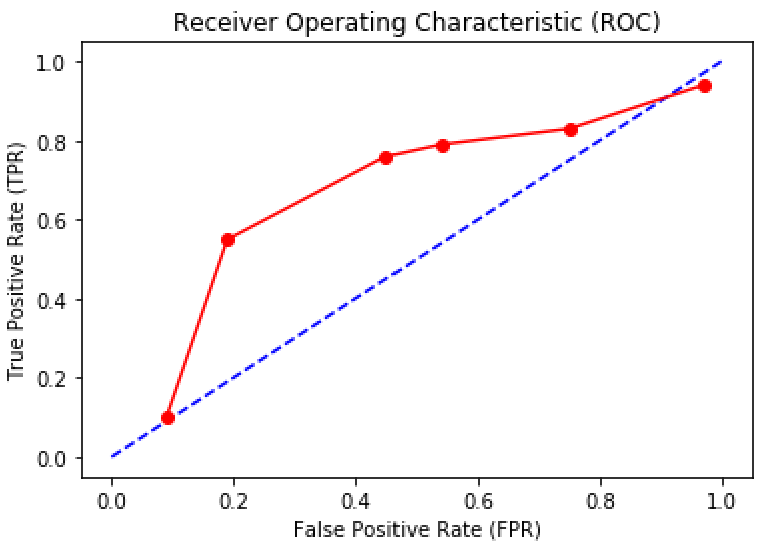


Figure 6 - ROC-AUC for the blue color histogram

The performance result was an AUC of 0.637 which is also better than random selection. It might be suitable for pre-selection of images but was not at a sufficiently accurate level to credibly detect a fake. As in the previous example, the level of performance for a classifier would be comparable to many people's ability at an art gallery.

Exploratory Visualisation - Use of ORB for feature extraction.

Feature extraction and grouping methods have been used for image classification including t-SNE [17] and SVM [18].

Further clues about Whiteley's style including the use of sweeping brush strokes, abstract lines, and subjects including nudes, birds & waterscapes were considered.

A further model was thus developed prior to the development of a full CNN-based classifier.

A classifier concept was developed relating to feature extraction similarity. The hypothesis was formed. "what if Whiteley's paintings contain similar features, which are sufficiently different from features painted by other artists?". If this were indeed the case, a unique classifier could be built around feature extraction. This classifier would have the potential for efficient computation and accuracy.

It was decided to try an alternative approach to feature extraction-based classification.

Several feature extraction methods are available including SiFT [19], SURF [20], and ORB [21]. Typical applications for these models include object detection and comparison between old and new images subjected to rotation, scale and translation.

ORB was proposed as a feature extractor for the following reasons:

1. ORB is open source and available for use in python
2. fast and efficient
3. built on the FAST keypoint detector [22] and BRIEF [23] descriptor
4. ORB has an orientation component which might help with discrimination due to fixed image pose between classes but different artistic / painting orientation

ORB creates a series of keypoints (vectors of feature similarity) between two images. A simple model was developed for classification as follows:

1. Shuffle through images in the same class
2. perform brute-force ORB feature extraction (keypoints)
3. select the highest ranked keypoint (most salient feature between images)
4. This keypoint pair forms a vector with magnitude and angle
5. Add the length of this vector to the vector of all images in the class (Euclidian length vector)
6. Create an angle distribution of this vector
7. If the features in each class are sufficiently similar to each other, yet different to the other class then a classifier can be created based on the feature extraction.

This method was executed on images from both classes in the dataset. As you can see from the Figure below, although the algorithm does an effective job at matching features of similar information content, these are not likely features with the same intent (nose and a nose, beak and a beak, window and a corner etc.). There was also insufficient discrimination of the vector set between image classes for a classifier to be created. Thus, it was decided to move to a more sophisticated model in the form of a CNN.

Improvements to the ORB extractor-classifier method are described in a later section.

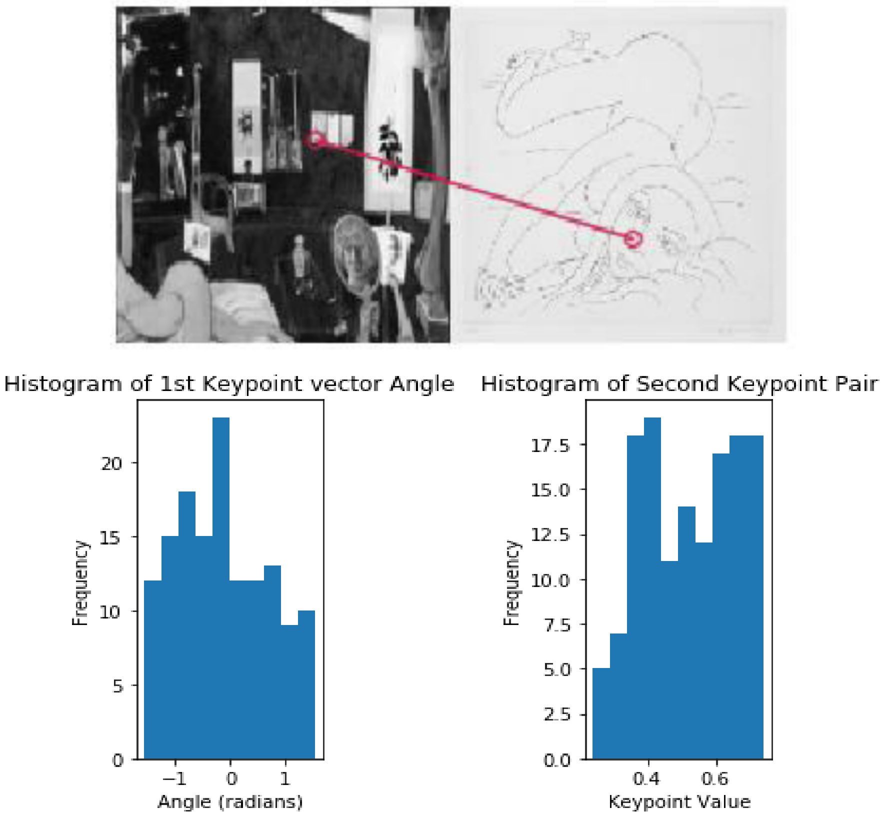


Figure 7 - ORB feature extraction on dataset including images (Whiteley class)

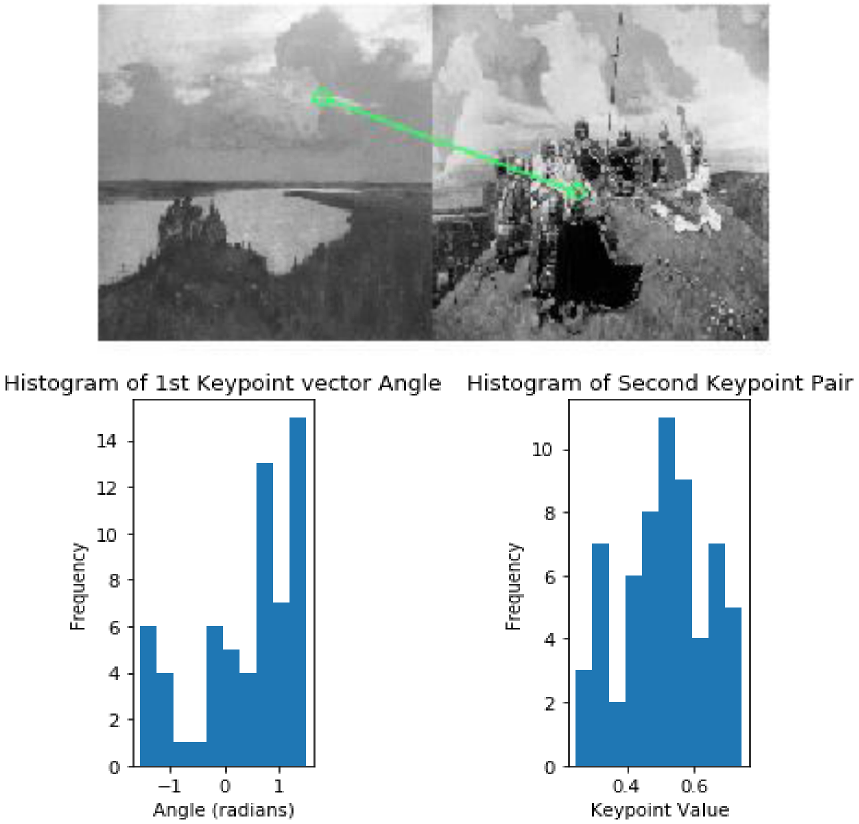


Figure 8 - ORB feature extraction on dataset including images (non-Whiteley class)

Two methods were employed to increase the classification accuracy of the images:

1) A CNN designed and trained from scratch
2) A CNN Transfer learning model trained from the ImageNet database

CNN built from scratch

The CNN built from scratch is developed from model architectures described in the literature [14], [15], [24]. The model architecture is defined in the table overleaf.

Scratch-built CNN - Algorithms and Techniques

Some of the important aspects of the model architecture:

A. Convolutional layers of input image size for feature extraction
B. Activation layers (relu equation) for high efficiency non-linearity and helps avoiding the vanishing gradient problem
C. Maxpooling layers – filter and stride of same length for dimensionality reduction / downsampling
D. Flattening layer to convert image array pooling data into a single dimensionality array for processing in the binary classification (activation layer)
E. Dense layers after filtering (convolution) maxpooling (downsampling) and flattening. The layers are dense because every node in the layer is connected to every node in the preceding layer. The objective of the dense layer is to help with classification
F. Dropout layers “drop out” a random set of activations in the layer by setting some fraction of weights to zero. This helps avoid overfitting and the problem of vanishing gradients.
G. Finally, a “sigmoid” activation to perform the classification (on two classes).

Scratch-built CNN - Description

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 254, 254, 16)	448
conv2d_2 (Conv2D)	(None, 252, 252, 16)	2320
activation_1 (Activation)	(None, 252, 252, 16)	0
max_pooling2d_1 (MaxPooling2	(None, 126, 126, 16)	0
conv2d_3 (Conv2D)	(None, 124, 124, 16)	2320
activation_2 (Activation)	(None, 124, 124, 16)	0
max_pooling2d_2 (MaxPooling2	(None, 62, 62, 16)	0
conv2d_4 (Conv2D)	(None, 60, 60, 32)	4640
activation_3 (Activation)	(None, 60, 60, 32)	0
max_pooling2d_3 (MaxPooling2	(None, 30, 30, 32)	0
conv2d_5 (Conv2D)	(None, 28, 28, 128)	36992
activation_4 (Activation)	(None, 28, 28, 128)	0
max_pooling2d_4 (MaxPooling2	(None, 14, 14, 128)	0
conv2d_6 (Conv2D)	(None, 12, 12, 256)	295168
activation_5 (Activation)	(None, 12, 12, 256)	0
max_pooling2d_5 (MaxPooling2	(None, 6, 6, 256)	0
conv2d_7 (Conv2D)	(None, 4, 4, 32)	73760
activation_6 (Activation)	(None, 4, 4, 32)	0
max_pooling2d_6 (MaxPooling2	(None, 2, 2, 32)	0
flatten_1 (Flatten)	(None, 128)	0
dense_1 (Dense)	(None, 32)	4128
activation_7 (Activation)	(None, 32)	0
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 1)	33
activation_8 (Activation)	(None, 1)	0

Total params: 419,809
Trainable params: 419,809
Non-trainable params: 0

Table 1 – Scratch-Built and Trained CNN for Binary Image Classification – Model Architecture

Scratch-built CNN – Limitations & Refinements

Several architectures were attempted before this architecture was selected.
Early issues and solutions included the following:

Issue
CNN training became "stuck" at low value or NaN
Resolution
Balanced the data sets (reduction in the number of non-Whiteley images)
Performed data augmentation
Increase the level of augmentation (number of images generated, translate, shear, flip, rotate)

Issue
CNN overfitting
Resolution
Add Dropout layer and increased level of Dropout
Reduced number of conv layers (CNN depth)
Reduced the size of the conv layers
Changed batch size

Issue
CNN now underfitting
Resolution
Updated the train-validate ratio (82/18%)
Increased the number of conv layers

Issue
Insufficient Accuracy
Resolution
Updated the optimiser to Adadelta
Used MSE as the loss function
Increased number of Epochs (Stable CNN)
Adjusted dropout ratio and conv size

Limitation
Insufficient number of images / data
Resolution
Develop pre-trained model and perform transfer learning

Scratch-built CNN - performance and benchmark

The final scratch-built CNN was able to achieve an accuracy of 85.8% after 40 epochs. There was little difference between train and test accuracy at 40 epochs showing that the model was optimally fitted. This accuracy is of similar performance to other models in the literature for artist classification given similar epoch and number of training images [15], [24].

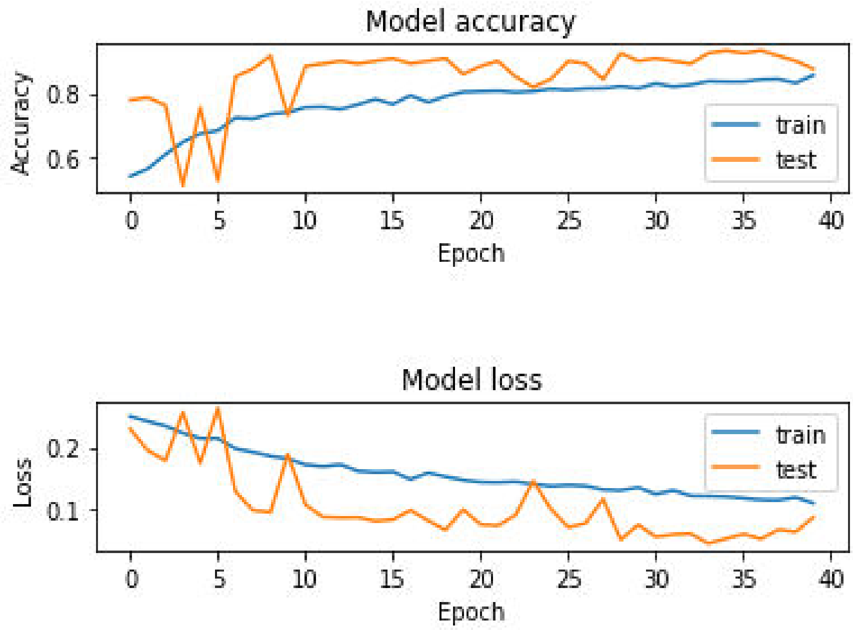


Figure 9 – Accuracy and Loss against Epoch (CNN built and trained from scratch)

Transfer Learning with pre-trained model

Transfer Learning - Algorithms and Techniques

The second approach was to make use of Transfer Learning techniques, to use an existing trained model and retrain on the image dataset.

Transfer learning is a machine learning method whereby the weights and model architecture from a previously trained model in a similar problem domain are truncated (tail). Then a new model is added to the existing model (head) along with the existing features (bottleneck features).

This new model is then trained on the dataset to be classified.

Transfer learning applies the previous learnings on objects (images) from previously trained, typically much larger datasets.

The advantage of the pre-trained model is that the Imagenet dataset which it was trained upon is massive (~14M images)

The disadvantage of the pre-trained model is the Imagenet dataset which it was trained upon had object classes different to the "style" features representative of art or artworks.

Transfer Learning - Description

Flatten	--
Dense	64
Activation	ReLU
Dropout	0.5
Dense	2
Activation	sigmoid

Table 2 –Transfer Learning Head Layer – Model architecture

the model architecture was VGG16 [12] which has been shown to have good accuracy and performance, as well as a relatively shallow layer structure which is suited to smaller datasets.

The head layer is depicted in Table 2 and consists of ReLU activation and Dropout layer as well as sigmoid activation for the final classification. Categorical cross-entropy was chosen as the loss function

Limitations and Refinements

Initially the model was overfitting, the addition of the Dropout layer assisted to reduce with the reduction of overfitting as well as reducing the batch size.

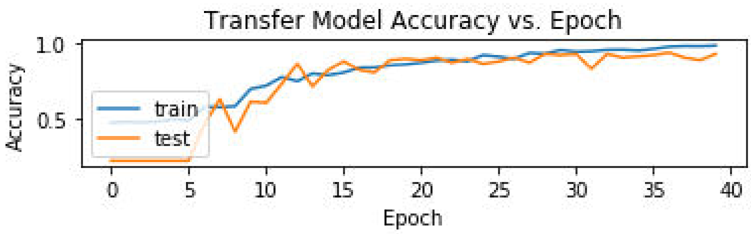
Performance was increased by increasing the dense layer and updating the Dropout rate.

Increasing the layer size (eg. Adding conv layers) tended to result in overfitting.

One key limitation of the transfer learning approach is that the Imagenet database contains classes of images such as dogs and cats rather than artistic "styles" of painting / painters.

Transfer Learning - performance and benchmark

The transfer learning model was able to achieve a final model accuracy of 92.6% after 40 epochs. There was low difference between train and test accuracy after 20 epochs showing that the model was optimally fit (no evidence of over / underfitting). This is similar performance to current research benchmarks using different methods - comparing Van Gogh to Picasso [15], [24] given equivalent number of images and epochs.



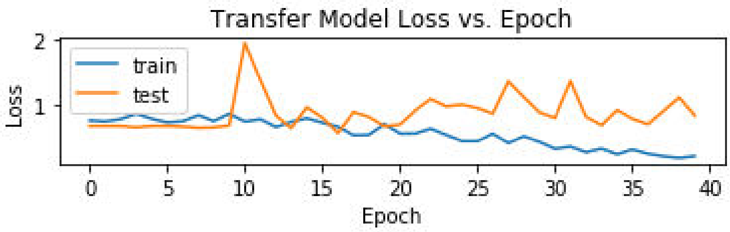


Figure 10 - Transfer learning model Accuracy and loss vs. Epoch

The Big Reveal - Is it a fake?

The evaluation of the alleged fake painting was then performed using the transfer model classifier. The model successfully classified the Whiteley and non-Whiteley paintings into the correct class.

The alleged fake was then piped into the model and the classifier predicted it to be........

A Whiteley.

Conclusion

This project and report set out to determine the possibility of using machine learning methods for style detection and fake painting detection - specifically for the artist Brett Whiteley.

The dataset was collected, curated and processed. The data was explored and visualised using techniques such as histogram and color extraction. Feature extraction for exploration and analysis was performed using the ORB algorithm.

Further two CNNs were developed, one a model built from scratch and the second using transfer learning of the VGG16 model pre-trained on the ImageNet database. While classification accuracies around 90% were achieved which is close to equivalent published performance, the classifier was unable to determine the fake painting as being distinct from the purported artist. So maybe Brett Whiteley did paint it and it was not a fake after all......

Improvement

There are many areas for improvement and future work. Some key areas for future improvements include:

<u>Data Curation</u>

Improving the quantity and quality of the ground truth images would assist with the classification performance. For example, taking hi-resolution photos with identical pose, scale, lighting and photographic parameters would result in a performance improvement. Identifying more paintings by Whiteley would also improve the performance.

<u>Visualisation - salient features</u>

An ensemble of more features could be created, and supervised learning techniques could be used to enhance the performance of the classifier. Some examples would be histogram peak, color vectors, spectral content and edge number for salient features and these feature values could be then included into a classifier (SVM / random forest).

<u>Visualisation - ORB</u>

While ORB was effective at determining areas of similar features, these features were not congruent with artistic features as observed by a human. Some suggested improvements to the ORB feature extraction model would include:

- increased number of keypoint pairs to form a multivariate classifier
- use of the cosine-squared loss function
- data augmentation (rotation and translation of each image per class)

<u>Scratch-built CNN</u>

Improvements would include increasing the number of layers and adjustment of the dropout rate for increased performance.

<u>Transfer Learning CNN</u>

VGG16 had the highest performance compared to NAS and Xception in this application. Trying different pre-trained models (Inception...) would be an example of further work. Slicing the bottleneck features earlier in the CNN (edge detection) may also assist.

If it were possible to perform transfer learning on a model previously trained on artwork, the performance would be potentially higher as the CNN would have obtained features based on artwork rather than everyday objects.

Reflection

A lot was learned during the course and during the completion of the Capstone project. Some highlights and observations:

1. It was interesting that some "low-tech" methods like image processing could produce a classifier with low computational load and performance that wasn't terrible. For embedded platforms with low performance requirements this may be a more effective approach than unsupervised learning methods.

2. Increased understanding of the difficulties and intricacies of developing a CNN. There were many areas to understand and investigate including:

    - layer number optimisation
    - input data balancing
    - loss function optimisation and selection
    - avoiding overfitting (the enemy!) in the presence of sparse data
    - parameter optimisation (dropout rate, image size, batch size)

3. Increased knowledge in cloud computing and increased skills in python. Some highlights were development of a GitHub account, setting up a jupyter notebook running on a cloud-based GPU instance on AWS, and learning more about some of the packages like CV2 and keras.

References

[1] http://adb.anu.edu.au/biography/whiteley-brett-17651 “Whiteley, Brett (1939-1992)” by Barry Pearce 2016.

[2] https://www.theaustralian.com.au/arts/brett-whiteley-forgery-case-to-go-to-trial/news-story/

[3] https://www.sbs.com.au/news/court-unable-to-determine-whether-whiteley-art-is-forged

[4] https://cloud.google.com/blog/products/ai-machine-learning/who-street-artist-building-graffiti-artist-classifier-using-automl

[5] Krizhevsky, Alex & Sutskever, Ilya & E. Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.

[6] Hussain, Mahbub & Bird, Jordan & Faria, Diego. (2018). A Study on CNN Transfer Learning for Image Classification.

[7] Gatys, Leon & Ecker, Alexander & Bethge, Matthias. (2015). A Neural Algorithm of Artistic Style. arXiv. 10.1167/16.12.326.v

[8] https://medium.com/mlreview/making-ai-art-with-style-transfer-using-keras

[9] https://www.analyticsvidhya.com/blog/2016/10/investigation-on-handling-structured-imbalanced-datasets-with-deep-learning/

[10] https://www.menziesartbrands.com/blog-post/brett-whiteley-arkie-under-shower

[11] https://www.theguardian.com/culture/2015/oct/26/brett-whiteley-influences

[12] “Very Deep Convolutional Networks for Large-Scale Image Recognition”, Karen Simonyan, Andrew Zisserman, University of Oxford, 2015, ICLR 2015

[13]http://openaccess.thecvf.com/content_cvpr_2017/papers/Chollet_Xception_Deep_Learning_CVPR_2017_paper.pdf

[14] https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html (Francois Chollet)

[15] http://cs229.stanford.edu/proj2010/BlessingWen-UsingMachineLearningForIdentificationOfArtPaintings.pdf

[16] https://en.wikipedia.org/wiki/Receiver_operating_characteristic

[17] https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1

[18] https://en.wikipedia.org/wiki/Support-vector_machine

[19] “Distinctive Image features from scale-invariant keypoints” David G Lowe, U British Columbia 2004.

[20] “SURF: Speeded-up robust features” Herbert Bay, Tinne Tuytelaars, Luc Van Gool, ETH Zurich 2008.

[21] “ORB: An efficient alternative to SIFT or SURF” Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski, Willow Garage, 2011 http://www.willowgarage.com/sites/default/files/orb_final.pdf

[22] “Machine learning for high-speed corner detection”, Edward Rosten and Tom Drummond 2010.

[23] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua, “BRIEF: Binary Robust Independent Elementary Features”, 11th European Conference on Computer Vision (ECCV), Heraklion, Crete. LNCS Springer, September 2010.

[24] http://cs229.stanford.edu/proj2018/poster/41.pdf