Image Forgery Detection

Gregory Hill and Emily Rager

Abstract

Mainstream photo editing applications have trivialized the alteration of digital images. Advancements in deep learning have further simplified both forgery and content creation such that any layman with the right equipment can participate. In an era of fake news, trust is a vital currency that can no longer be accepted. The intent of this project was to assess the suitability of deep learning for counterfeit image detection in an effort to target illicit content, but its implications extend much further. We developed two datasets using popular deep learning frameworks for training our models. The first was taken from a selection of paintings in which new artistic styles were automatically applied, then separated. Another smaller set was built which learned unique facial patters and swapped the faces of multiple subjects. Our developed model demonstrated an accuracy of over 90% in many of our trials on stylized content, with significant benefit from Error Level Analysis (ELA) pre-processing. Trials on our dataset with facial modifications were equally as successful, but would benefit from more data. Nevertheless, the results of this investigation conclude that Deep Neural Networks (DNNs) can successfully classify forged media through inherent strengths.

1. Introduction

Image manipulation is the art of re-shaping digital content to portray a skewed version of the truth. There are undoubtedly many benefits to altering media in this way, particularly in the movie industry, but 'fake' content can have dire consequences for those it misrepresents. For example, there was a great deal of controversy recently when an image of a prominent campaigner for gun control starting circulating on social media (Wright, 2018). It falsely portrayed the individual tearing up the US constitution in an effort to further a far-right agenda. This type of news is all too common in today's society, and there is little defence (Nature, 2017).

Digital forensics has played an important role over the last few decades but Garfinkel (2010) argues that the 'golden age' may be coming to an end. Failings are becoming increasingly present due to the ageing of once standard technologies which falter due to, for example, format incompatibilities. The author underpins his description of the 'golden age' by traits such as the widespread usage of Windows XP, specific hardware standards and the fact that there were comparatively fewer file formats of forensic interest. Fundamental advances in the technology used by the general populous include growing local and 'cloud' storage, prevalence of removable devices that are difficult to image and comprehensive encryption. It simply is not possible to assess digital evidence in the same way it once was. Adversarial techniques, such as those described by Gloe et al. (2007), further the need to build more resilient tools and countermeasures.

A recent trend has utilized artificial intelligence in a perverse manner. Users are downloading pre-built machine learning algorithms which can be trained to swap an individual's face in target content with that of another (Deepfakes, 2018) with veritable ease. It has garnered significant negative media attention however due its misrepresentation of notable actors in pornographic content (Lee, 2018). The indecent images will not be presented here, but the reader is directed to Figure 1 for a respective depiction of the content's realism. As the model was only trained for a few hours on a relatively average Graphics Processing Unit (GPU), the reality is that deep learning is no longer exclusive to researchers with specialized knowledge and far from limited to those that can afford it.

Deep learning provides the ability to generate realistic forgeries, but does its potential also extend to forgery detection? There is a clear need for new technologies which automate the forensic process, so the following report will outline efforts to afford such a tool built with a Convolutional Neural Network (CNN). Many researchers have previously tackled this problem, their endeavours are outlined in Section 2. The methodology (Section 3) is split into two components for two unique challenges; data-set creation is initially outlined, then we define a binary classifier which serves to discriminate between authentic images and their forged counterpart. The results from our solutions are listed in Section 4 followed by our final thoughts in Section 5.

(a) Original (b) Fake

Figure 1. DeepFakes Example

2. Related Work

Many techniques employed in contemporary literature were founded a number of years ago across several highly regarded digital forensics papers. Popescu & Farid (2004), for example, introduced many statistical techniques to detect re-sampling in digital forgeries. Typically, re-sampling is a mathematical technique used to scale an image (Sachs, 2001), but in this case they assessed its use in splicing - the process of cutting one image into another (Ng & Chang, 2004). This is essential because when two images are spliced together, at least one has to be re-sampled in order to maintain consistency. Popescu & Farid (2004) discovered that re-sampling introduces "specific correlations between neighbouring image pixels" and identified a comprehensive and robust procedure for detection. Many other researchers have had similar success in splicing detection, Hsu & Chang (2006) demonstrated a reasonably high accuracy in their 'semi-automatic' binary classifier by computing geometric invariants based on observations by Ng et al. (2005) on authenticity properties of legitimate photographs. There were two specific identifiers of an original picture: natural scene quality, relating to lighting consistency and reflective patterns, and natural imaging quality, which indicates that the image must have been captured through some 'image acquisition device'.

The Joint Photographic Experts Group (JPEG) developed their international compression standard over two decades ago as part of the International Standardization Organization (ISO) and International Electrotechnical Commission (IEC) (Wallace, 1992). It is still commonplace to this day and can reduce an image to between $1/10^{th}$ and $1/50^{th}$ of its original size. Forensic analysts exploit this scheme to identify differing compression rates, as most cameras save straight to JPEG, a second round of compression might suggest that it has been re-compressed after alteration in a digital editing toolkit. Popescu & Farid (2004) leveraged specific correlations in the image transformation process to distinguish between single and double compression with high accuracy, though there are now far more advanced techniques. Luo et al. (2010) presented a theoretical analysis on the study of JPEG images with regard to error and compression rates. One approach was able to distinguish whether a bitmap had previously been compressed under the JPEG scheme, with high accuracy. Similar work was applied later by Krawetz (2012) in what is now known as ELA. With repeated lossy compression, it is possible to study a difference in the level of compression artefacts. The image is subjected to an additional round of lossy compression and the result is subtracted from the original data. Resulting artefacts may be visually assessed for authenticity which has proven controversial. Specialists have debunked claims that this type of analysis is effective (Steadman, 2013) as it relies too heavily on subjective visual analysis, as opposed to a more rigorous mathematical treatment.

Contemporary literature on image forgery detection has similarly explored deep learning. Bharati et al. (2016) was able to detect facial re-touching with over 87% accuracy on their custom dataset of 330 images using a novel Boltzmann machine algorithm. They also claim a particularly high accuracy of 99% on datasets from comparable literature on 'make-up' detection. It is not clear why the margin changed quite so substantially, but the authors poise that the problem is comparatively easier. A recent paper by Bunk et al. (2017) vastly improved on these results to localize manipulated regions in images. They experimented with CNNs and Long Short-Term Memory (LSTM) systems to isolate re-sampled features (Popescu & Farid, 2004) for patch classification. The final classification accuracy of their proposed method was stated to be ~94%. Mohammed et al. (2018) provide an additional technique to boost the efforts of deep learning classifiers. The authors defined a 'copy-move' algorithm that increased accuracy in their tests by around 8-10%. This is effectively the same as splicing (defined earlier), with similar re-sampling features, but they aim it to be a pre-filtering step. It is not exactly clear if this would prove to be beneficial in the other papers discussed here.

3. Methodology

The undertaken project has been divided into two sections, a preliminary Proof of Concept (PoC) investigation on the feasibility of our research question (as discussed at the end of Section 1) was completed prior to commencing the latter study. This second aspect, which is outlined herein relates to the further exploration of the research question. Given that the initial PoC showed favourable results both with and without pre-processing with ELA, a decision was made to further split the methodology of the current phase into two. Further testing of a binary classifier using stylised images, for which a more comprehensive dataset was generated. Moreover, we aimed to test our most successful model with another dataset comprising face-swapped actors. Subsection 3.1 discusses the creation of our datasets, and provides examples as well as dimensions for both the Neural Style and Facial Alteration datasets. The following Subsection 3.2 defines the CNN structure utilized.

3.1. Dataset Creation

This section will first outline the dataset creation as an extension of the preliminary method utilized in our prior paper. It will then explore the creation of a novel development dataset not previously considered.

3.1.1. Neural Styles

With the aid of an open-source neural-style transfer tool (Smith, 2016), we transformed part of the 'Painter-by-Numbers' (Nichol, 2016) dataset. Built for an online competition of the same name, the original aim was to construct a classifier which could determine a similarity score be-

tween paintings to assess if they were made by the same artist. It contains 103250 unique paintings split into a training set of 79433 and a test set of 23817. For the purposes of our project, we chose to utilize a subset of the full dataset, namely the `train_2` subset. This set merely contains 8476 unique paintings from a variety of authors. From this, 100 paintings were removed to be used as source style images. The intent was to use a mixture of the variety of styles as well as 7 hyper parameter configurations to stylize the remainder. However, due to issues with getting the required software working as intended on the provided GPU cluster, only 848 resultant stylized paintings were produced. The hyper-parameters are based on the default settings provided by (Smith, 2016).

The full list of parameter configurations used is as follows:

1. Default
2. Original Colours
3. `style_weight`=1e2
4. `style_weight`=1e9 and original colours
5. `content_weight`=4e1 and `max_iter`=50
6. `content_weight`=1e4, `style_weight`=5, and `max_iter`=2000
7. `content_weight`=1e4, `style_weight`=5, and original colours

It was therefore decided to use an unbalanced dataset, with the full 8376 original paintings and 848 stylised paintings for a near 10 : 1 ratio. Furthermore, for validation purposes hold-out validation was used, and the set subdivided into training and testing sets with a 80 : 20 split.

Figure 2. Scene with a Windmill - James Webb

For the purposes of applying a Neural Style to the paintings, we used the neural style generation tool by Smith (2016) which can alter the content and style weights for very unique outputs. It is also possible to keep the original colours and update the processing time for a lighter or heavier feel. The Starry Night by Vincent van Gogh is well known for it's enchanting color palette and expressive swirls. To demonstrate the tools applicability we trained on the unique style and applied it to painting from the dataset - Figure 2. The output, shown in Figure 3, is quite strong and adequately captures the original style. Not every image in our resultant dataset is quite to vivid however. Each image was generated with a random configuration from the list above, and all of the styles were also chosen at random, hence some images may appear to be identical to their original. What is important however, is that the neural style transfer process will inject some form of artefact in each image.

Figure 3. "Scene with a Windmill" with Starry Night Neural Style

3.1.2. Facial Alteration

We utilized a popular face swapping application (Deepfakes, 2018) based on the original Reddit thread of the same name to transform a set of collected images. Our scraping tool fetched the top one hundred results from a given Google query and downloaded all images into a set folder. This process was repeated ~a dozen times to gather relevant images of a particular subject. As in Figure 1, this was initially trialled with images of the two notable actors Nicholas Cage and Patrick Stewart - with unique, discernible, facial geometry. These directories were then manually pruned for inconsistencies, such as watermarks or incorrect subjects, and copied into the source directory of the face swap tool (Deepfakes, 2018). The algorithm further cleaned the data by aligning and cropping all faces before commencing training.

3.1.3. ELA

ELA, as briefly summarized in Section 2, is a technique for assessing the differing error levels throughout an image (Krawetz, 2012). Significantly brighter noise is a strong indicator of digital manipulation. For example, the ELA of Figure 2, shown in Figure 4 is almost black which suggests that it was only compressed once. However, the stylized painting in Figure 3 paints a completely different picture in Figure 5. The course treatment of our earlier conversion clearly altered the error rates, as the image shows significant distortion compared to that of Figure 3. This is a good example of the benefit to this analysis, but not all images are affected quite so dramatically.

Figure 4. Scene with a Windmill - ELA

Based upon the findings from the interim report, namely that ELA pre-processing of the dataset has a significant improvement in both accuracy and generalisation of the classifier, we decided to copy the dataset produced in Section 3.1 and perform ELA on every single image in the set. This will be used as a comparison metric, as well as against the baseline provided earlier. See Section 4 for a tentative discussion of the results.

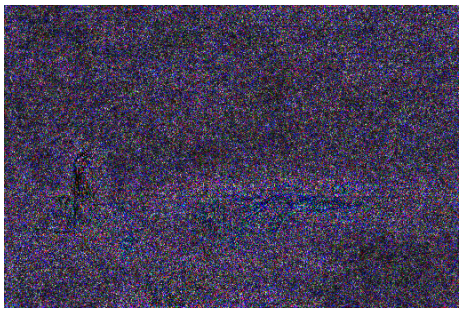

Figure 5. Stylized Painting - ELA

3.2. Classification

After experimenting with several simplistic classifiers such as logistic regression and k-nearest neighbours we settled on the binary classifier proposed in our initial design. The CNN utilizes Keras (Chollet, 2015), the Python deep-learning library, to further extend upon TensorFlow and to enable rapid prototyping in addition to GPU acceleration. Further analysis verified the optimal layers for this task; for example, the Rectified Linear Unit (ReLU) activation function was chosen because of its exceptional performance in similar tasks (Simonyan & Zisserman, 2014). For the purposes of this project, a simple 6-layer sequential (linear) classifier was built with the following structure:

1. 2D Convolution w/ ReLU
2. 2D Max Pooling
3. 2D Convolution w/ ReLU
4. 2D Max Pooling
5. Flattening Layer
6. Dense (128 Units) w/ ReLU
7. Dense (1 Unit) w/ Sigmoid

Convolutional networks (LeCun et al., 1995) combine shared weights, local receptive fields and spatial subsampling to automatically extract useful features for successful classification. A small filter (kernel) is scanned over the input image to reduce the dimensionality and output a smaller matrix of pixel values. The max pooling layers are used to further reduce the space into a more manageable size by taking the maximum value in each 2×2 filter. The last two layers are referred to as "Dense" in the Keras documentation, but this is synonymous with the 'fully-connected' layer. This simply means that every node from the previous layer is connected to every neuron in the next. So, the two-dimensional output from the last max pooling layer is essentially flattened to one-dimension for it to learn a (possibly non-linear) function in its space. The final layer in the model serves to commit the binary classification on its input.

4. Experiments

4.1. Neural Style

Once the initial PoC dataset had been produced for the first deadline of the project, we tested it with a baseline binary classifier as described in Subsection 3.2. While this original test admittedly did produce very good validation accuracy and respectable loss for the first half of the epochs, the dataset used only had a single neural style to train against, resulting in over-fitting and poor generalization. For the purposes of illustration, training was performed past the local loss minima, and the result can be observed in Figure 6.

As a result of this discovery, it was decided to produce a further extended dataset comprised of 100 different neural styles, in addition to multiple parameter configurations for each of those, resulting in 700 possible style applications.

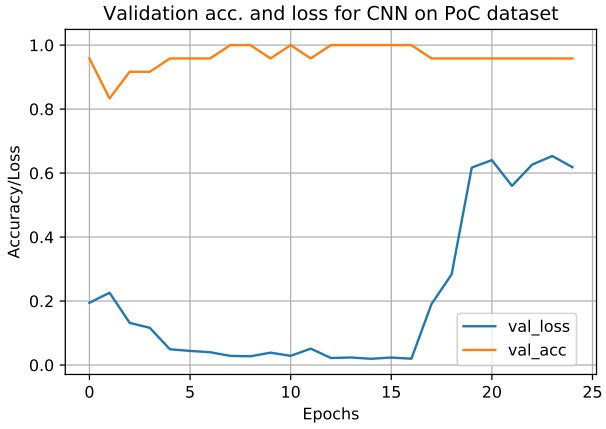

Figure 6. Baseline test of CNN on PoC dataset.

Once the new and improved dataset had reached an acceptable size, experiments were carried out in a similar fashion as the baseline seen in Figure 6. However, some slight changes were made to batch sizes and step counts to support the increased and unbalanced dataset's narra-

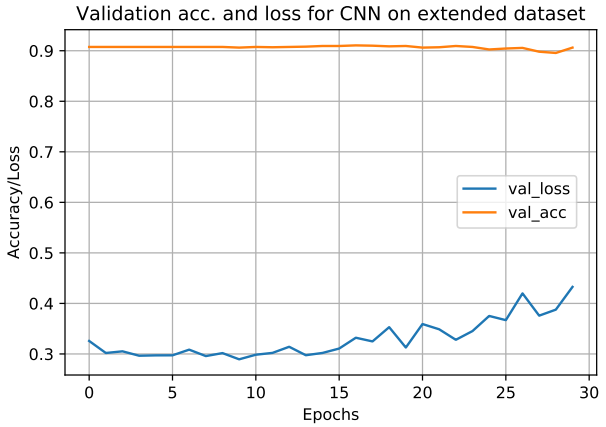

Figure 7. CNN performance on the extended dataset.

tive. These changes were primarily made in order to ensure regular check-pointing, such that if any problem was encountered that would affect the remaining run, we would be able to resume from a saved state rather than from scratch.

The results clearly show that despite having 700 possible neural style combinations that could be applied, general accuracy is very good at ~ 91% on the validation set - see Figure 7. While the classifier certainly seems to suffer from local loss minima, it is less pronounced than on the PoC dataset.

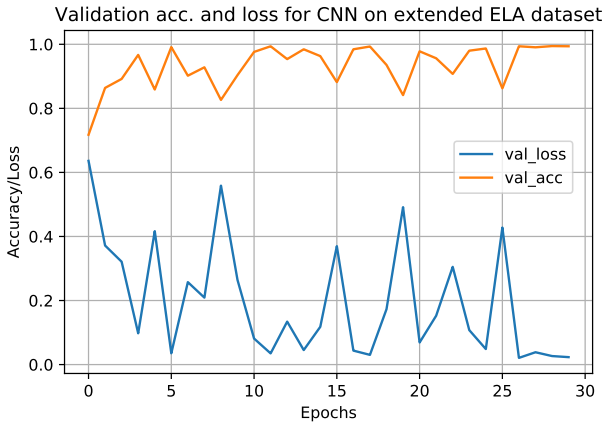

Figure 8. CNN performance on the extended dataset w. ELA pre-processing.

Following on from the prior experimentation with ELA pre-processing of the dataset, the CNN was trained against an ELA edition of the extended dataset, which the exact same configuration as the last experiment. While the pre-processed dataset appears to converge at a local loss minima faster than the other models, it has many maxima and minima - see Figure 8. Given these irregularities which do not seem to converge toward any specific trend, it is likely that while the pre-processing aids in training, it also hinders the model if there are too many distinct features as it simply highlights error prone areas (Luo et al., 2010).

4.2. Face Swap

Our final dataset for this final task comprised 307 × 2 (Real / Fake) images in the training set and a further 74 × 2 for testing. The 'DeepFakes' tool described in Section 3 was able to extract a selection of faces from a large collection of scraped images which left many outliers. Furthermore, the conversion method left a number of images untouched and we had to prune a greater number of images from the set than originally expected. This left us with 762 usable examples.

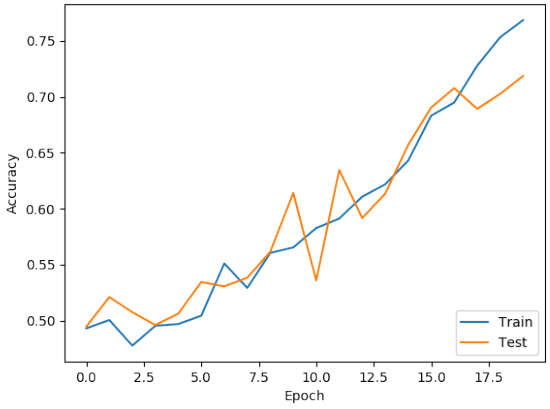

Figure 9. CNN accuracy on the face dataset.

Although the model would clearly benefit from a longer run, we can see its performance excels over time. Figures 9 and 10 illustrate the results from one test run. It concludes in this with an accuracy of ~ 80% on the training data and ~ 70% on the validation. Its respective loss plateaus till around the tenth epoch and continues to slowly degrade, which shows significant progress.

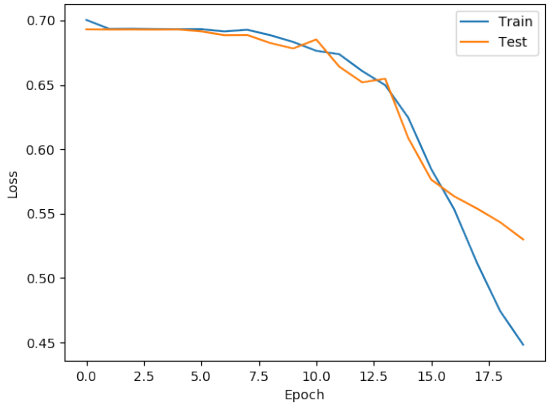

Figure 10. CNN loss on the face dataset.

As in the previous task, the next step was to experiment with the ELA transformed dataset. Due to incompatibilities with the process, the dataset had to be cut further as it contained several Portable Network Graphics (PNGs) and a number of grayscale images.

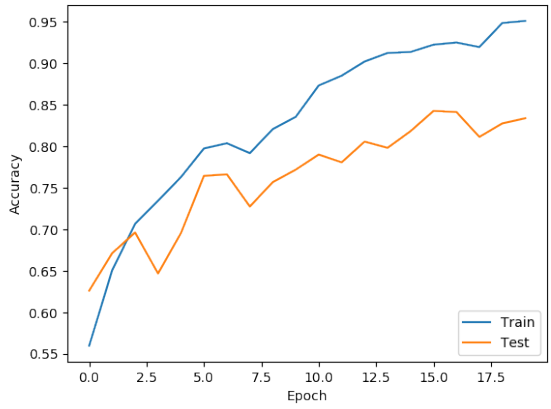

Figure 11. CNN accuracy on the face dataset.

The results shown in Figure 11 are unlike those found in Subsection 4.1. Training accuracy is ~ 95% and validation peaks at 0.8425. This is significantly greater than the accuracy shown pre-ELA in Figure 9.

5. Conclusions

This paper has highlighted an approach to image forgery detection that adequately discriminates between original and modified images. Two new datasets were introduced due to the noted lack of available material with a clear separation between real and fake. Our CNN demonstrates that there are substantially many features correlating to authentic and forged content even after applying ELA. In some cases, such as detecting stylised paintings, pre-processing the content with ELA even improves upon the performance of the classifier in question. Though data collection still requires manual input to sort the images, a trained model based on the given dataset should be sufficient for an inexperienced analyst to test a subject image for authenticity. Therefore, we believe the experiments undertaken to have addressed one of the major issues highlighted in Section 1 regarding the need for advanced intelligent tools as requested by Garfinkel (2010). Given the relatively simple classification model used and the good performance achieved, it is certainly a feasible area of research. Fundamentally however, deep learning can serve to fight in the battle against illicit media to quickly identify non-authentic content.

5.1. Risks evaluation

At the beginning of this project, a risk assessment was undertaken to evaluate which potential risks had a chance to affect the project in a major way. For reference, this list can be found in Table 1. In this section we intend to give a short breakdown of what problems were encountered and how they were resolved.

It is quite clear from both the preliminary work, as well as the results of the extended dataset that poor features and insignificant accuracy were not problems encountered.

Risk	Likelihood	Impact
Poor Features	0.1	10
Insignificant Accuracy	0.2	8
Dataset Size	0.1	8
Time Constraints	0.4	7
Set-up Breakdown	0.1	9
Travel & Illness	0.5	2

Table 1. Risk Assessment from initial project planning

However, the assumption that problems with regards to dataset size would have a low probability of occurring was far from the truth. In fact, this is where most of the issues have been. Despite this, however, good performance was achieved.

Perhaps the biggest issue encountered has been bad weather and strikes preventing meetings from happening, and finally illness for one of the group members had more of an impact than expected.

5.2. Future Work

Many of the experiments shown in Section 4 would benefit from substantially more data. Due to various unforeseen issues with the utilised packages we were not able to process the full intended dataset size, particularly in the latter challenge. In particular the production of the stylised dataset, as well as the face swapped datasets are resource intensive processes. While more data certainly would improve the possible avenues of classifier construction, it would likewise allow for different types of classifiers, such as elaborating on what kind of alterations stand out as ‘fake’.

Despite the controversy surrounding ELA highlighted in Section 2, we chose to utilise it in a number of our experiments. This is due to the fact that it significantly increased the accuracy in early tests and because we trialled several of the pre-processing techniques from the interim report without success. Specifically, Principal Components Analysis (PCA) did not provide enough granularity and Noise Analysis simply proved ineffective. Further work would be advantageous to reassess these techniques in our model, or identify more suited methodologies like that proposed by Mohammed et al. (2018) for better generalization.

References

Bharati, Aparna, Singh, Richa, Vatsa, Mayank, and Bowyer, Kevin W. Detecting facial retouching using supervised deep learning. *IEEE Transactions on Information Forensics and Security*, 11(9):1903–1913, 2016.

Bunk, Jason, Bappy, Jawadul H, Mohammed, Tajuddin Manhar, Nataraj, Lakshmanan, Flenner, Arjuna, Manjunath, BS, Chandrasekaran, Shivkumar, Roy-Chowdhury, Amit K, and Peterson, Lawrence. Detection and localization of image forgeries using resampling features and deep learning. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pp. 1881–1889. IEEE, 2017.

Chollet, FranÃğois. keras. https://github.com/fchollet/keras, 2015.

Deepfakes. deepfakes_faceswap. https://github.com/deepfakes/faceswap, 2018.

Garfinkel, Simson L. Digital forensics research: The next 10 years. *digital investigation*, 7:S64–S73, 2010.

Gloe, Thomas, Kirchner, Matthias, Winkler, Antje, and Böhme, Rainer. Can we trust digital image forensics? In *Proceedings of the 15th ACM international conference on Multimedia*, pp. 78–86. ACM, 2007.

Hsu, Yu-Feng and Chang, Shih-Fu. Detecting image splicing using geometry invariants and camera characteristics consistency. In *Multimedia and Expo, 2006 IEEE International Conference on*, pp. 549–552. IEEE, 2006.

Krawetz, Neal. Image forensics : Error level analysis, 2012. URL http://www.errorlevelanalysis.com/.

LeCun, Yann, Bengio, Yoshua, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

Lee, Dave. Deepfakes porn has serious consequences, 2018. URL http://www.bbc.co.uk/news/technology-42912529.

Luo, Weiqi, Huang, Jiwu, and Qiu, Guoping. Jpeg error analysis and its applications to digital image forensics. *IEEE Transactions on Information Forensics and Security*, 5(3):480–491, 2010.

Mohammed, Tajuddin Manhar, Bunk, Jason, Nataraj, Lakshmanan, Bappy, Jawadul H, Flenner, Arjuna, Manjunath, BS, Chandrasekaran, Shivkumar, Roy-Chowdhury, Amit K, and Peterson, Lawrence. Boosting image forgery detection using resampling detection and copy-move analysis. *arXiv preprint arXiv:1802.03154*, 2018.

Nature. Image doctoring must be halted. *Nature*, 546 (7660):575–575, June 2017. doi: 10.1038/546575a. URL https://doi.org/10.1038/546575a.

Ng, T-T and Chang, S-F. A model for image splicing. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 2, pp. 1169–1172. IEEE, 2004.

Ng, Tian-Tsong, Chang, Shih-Fu, Hsu, Jessie, Xie, Lexing, and Tsui, Mao-Pei. Physics-motivated features for distinguishing photographic images and computer graphics. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pp. 239–248. ACM, 2005.

Nichol, Kiri. Painter by numbers, 2016. URL https://www.kaggle.com/c/painter-by-numbers.

Popescu, Alin C and Farid, Hany. Statistical tools for digital forensics. In *International Workshop on Information Hiding*, pp. 128–147. Springer, 2004.

Sachs, Jonathan. Image resampling, 2001. URL http://www.dl-c.com/Temp/downloads/Whitepapers/Resampling.pdf.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Smith, Cameron. neural-style-tf. https://github.com/cysmith/neural-style-tf, 2016.

Steadman, Ian. 'fake' world press photo isn't fake, is lesson in need for forensic restraint, 2013. URL https://www.wired.co.uk/article/photo-faking-controversy.

Wallace, Gregory K. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38 (1):xviii–xxxiv, 1992.

Wright, Mike. Fake images of parkland shooting survivor, 2018. URL https://www.telegraph.co.uk/news/2018/03/26/fake-images-parkland-shooting-survivor-emma-gonzalez-tearing/.