Morgan Gareth Skillicorn
May 2025
Abertay University
Student Number: 1903578
Course: MSc Technical Art and VFX 2024/25
Module: Artificial Intelligence and Machine Learning CMP11 Unit 1
Module Leader: Javad Zarrin

Using a Convolutional Neural Network (CNN) to Classify Images Containing Spatial Aliasing Artefacts (Moiré)

ABSTRACT

An exploratory look at the use of convolutional neural networks to classify images containing moiré artefacts or aliasing. Moiré, a type of large-scale interference pattern poses a particular issue in the virtual production industry where digital cameras are used in conjunction with LED panels to produce a final image, in-camera, often referred to as in-camera VFX (ICVFX). Commonly referred to as moiré artefacts, spatial aliasing is a phenomenon that occurs when reconstructed signals from samples contain low frequency components that are not present in the original sample. This is of particular concern in virtual production as the resolutions of LED volumes often surpasses the resolution capturable by the camera sensor. Due to the issues caused by moiré patterns it becomes an important task to monitor the occurrence of moiré on set and adjust the shoot to mitigate it. Brompton include several solutions in their LED processor technology including genlock and the ability to add a phase offset to the panels, however often it is easier to simply shift focus on the camera (Brompton Technology, 2025). Although there are a variety of quick fixes on set, nothing can be done to solve the issue if it is not noticed. This project explores the use of CNNs for early detection of moiré to alert on-set crew to potential issues. With the resources available and limited training time a result of 92% accuracy was achieved at classifying moiré and clean images correctly using a dataset of 10,000 4k resolution images. Several considerations had to be made when preparing the training dataset mainly what manipulations could be done to produce data of a manageable size yet still preserve the moiré interference patterns. Due to the overall size of the dataset a specialised compute server containing two Nvidia H100 graphics cards was utilised training the most accurate model over 20 epochs in approximately 86 hours.

INTRODUCTION

Commonly referred to as moiré artefacts in the virtual production industry, spatial aliasing is a phenomenon that occurs when reconstructed signals from samples contain low frequency components that are not present in the original sample. In photography this commonly occurs when a high frequency pattern such as fabric or brickwork exists. (Steve Marschner, 2016)

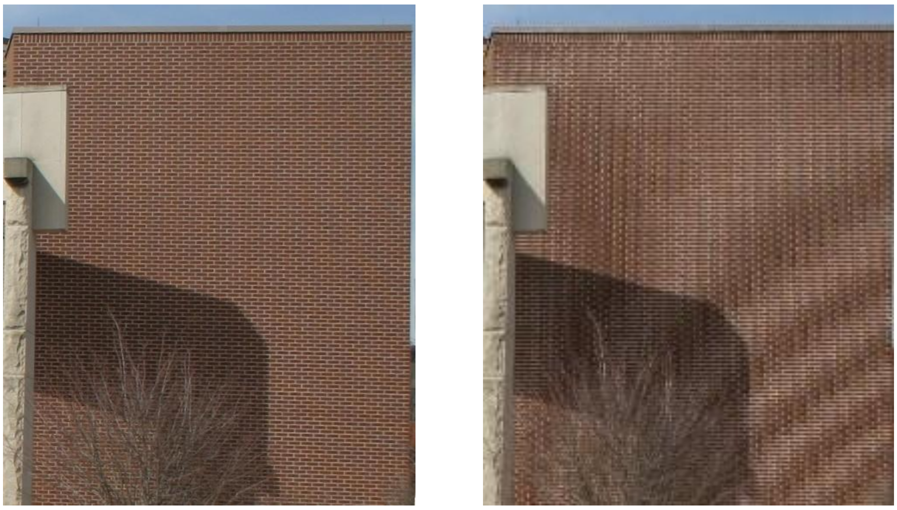

Figure 1. A photograph of a brick wall when badly down sampled introducing the appearance of special aliasing or moiré artefacts (Maksim, 2005)

The use of LED panels with high frequency LED arrays along with digital cinema cameras makes spatial aliasing an issue that must be considered and monitored much more closely than it has been previously. The introduction of aliasing in footage shot on set is a huge problem that can render entire takes unusable and cost studios significantly in VFX spending to remove the interference patterns or even require entire re-shoots for problematic takes.

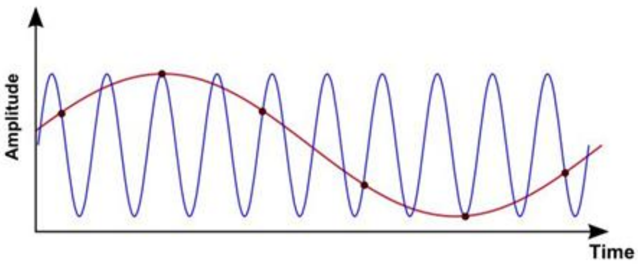

Figure 2. A sine wave sampled at a rate the does not recrate an accurate depiction of the sampled signal (Data Physics, 2016)

A common method implemented on set to reduce the occurrence of spatial aliasing is to shift the focus away from the high frequency pattern created by the LED array, reducing the number of high frequency elements and allowing the lower sample rate of the camera to more accurately recreate the image. Speaking to several professionals in the virtual production industry; a rule of thumb that the focal point should stay at least one metre away from the LED panel array for each millimetre of pixel pitch is adhered to reducing the occurrence of moiré. The filming of an LED panel with a digital camera could be considered a form of dynamic signal analysis making the application of several signal analysis principals relevant in this area of study (Zhuge, 2024).

It is also worth noting that temporal aliasing poses a problem on a virtual production shoot. This tends to occur when the camera's frame rate is out of sync or too low to capture high frequency changes on an LED wall resulting in banding or flickering effects (Netflix, 2025). This problem is much more straightforward to solve with genlock synchronization systems and cameras with global shutters. Classifying temporal and spatial aliasing patterns separately would be an interesting area of study however without a large custom dataset it falls outside the scope of this project.

It states in Real-Time Rendering (Tomas Akenine-Möller, 2008) that "The process of rendering images is inherently a sampling task." This same consideration should be made for digital video capture using LED volumes.

Figure 3. Spatial aliasing present when capturing an LED panel array (Rigard, 2025)

It is well documented that the application of a low pass filter to an input signal will remove visible aliasing artefacts (Zhuge, 2024). In this case the low pass filtering is done by moving the focal point away from the high frequency pattern created by the array of LEDs.

Nyquist-Shannon sampling theorem states that if a system uniformly samples an analogue signal at a rate that exceeds the signal's highest frequency component by a factor of at least two then the original analogue signal can be recovered exactly from the discrete samples (Shannon, 1998). When working with digital cameras the spatial sampling limit is based on the resolution, sensor size, lens and distance from the LED wall meaning it is difficult to directly increase the sampling rate in line with Nyquist-Shannon theorem.

This assortment of complex issues describes why the careful monitoring of the presence of aliasing must be happen on-set. This project does not intend to remove the existence of moiré patterns but instead detect it. This early detection allows the on-set crew to make informed decisions about the shot composition and the mitigation of unsightly aliasing artefacts.

PROBLEM STATEMENT

Background

LED walls and volumes are being used more and more in virtual production and as creative practitioners push the boundaries of this technology optical artefacts become a real issue. While understanding the causes of these artefacts, whether it be temporal aliasing or spatial aliasing, allows on-set technicians to solve most issues; identification of problematic footage is an issue that must be closely monitored.

Problem

Moiré patterns, flickering, rolling bars and aliasing are all optical artefacts that degrade image quality. These optical artefacts each have several corresponding solutions, assuming the issue is identified promptly.

Figure 4. Moiré aliasing present on an LED wall (Kadner, 2024)

Consequences

Optical artefacts degrade image quality, disrupt visual continuity and increase post-production workloads. Identification of the presence of optical artefacts allows film crews to reshoot and adjust the composition appropriately.

Project Goals

This project intends to explore the use of machine learning, specifically computer vision and convolutional neural networks to classify problematic images containing moiré artefacts. This model could then be used in conjunction with images sampled from a video feed on-set to identify moiré patterns, alerting the on-set crew. The ability to sample on-set data will not feature in this project and instead the possibility of using CNNs to classify problematic images will be explored.

RELATED WORK

It is no secret that the presence of aliasing has been widely researched in computer graphics. Many areas of this research are relevant in the virtual production industry. The work of Anjul Patney and Aaron Lefohn of Nvidia explores the use of a deep neural network in a similar fashion. “Detecting Aliasing Artifacts in Image Sequences Using Deep Neural Networks” (Patney & Lefohn, 2018) however is intended for use in fully rendered images. When working with these fully rendered images more flexibility exists in the creation of the dataset and the type of data that the model can be trained with. The input tensor size used by Patney and Lefohn is 64 × 64 × 4 RGB. This approach appears to work when identifying moiré in smaller samples but is unlikely to work when utilising a 4k image resolution without a custom dataset.

Eldho Abraham proposes the use of wavelet decomposition and a convolutional neural network to detect moiré patterns (Abraham, 2018) in order to prevent spoofing in sensitive applications such as biometric identification and optical character recognition systems. (Crowley, 2005) (Vetterli, et al., 2014). Common denoising approaches such as the application of a low pass filter to remove high frequency aliases are well documented (Mahya Hazavei & Reza Shahdoosti, 2017) however these methodologies can result in the removal of actual high frequency details in the image resulting in a loss of information, especially in an industry such as cinema.

The subject of aliasing with reference to convolutional neural networks is explored in the paper “How Convolutional Neural Networks Deal with Aliasing” (Ribeiro & Schön, 2021). The authors propose that images used in a CNN are progressively downscaled at each max-pooling layer posing the risk of the formation of moiré artefacts and aliases as high frequency components become indistinguishable from the low frequency components that comprise the image. This is a consideration that must be made when training a model to detect this type of spatial aliasing as the ground truth images could have aliasing introduced during the training process resulting in the model being trained on incorrect data. Zhang discusses the use of anti-aliasing low pass filters in a CNN as most modern networks don’t consider the issues that aliasing poses (Zhang, 2019) however doing so in this implementation could negatively impact the efficacy of the model at classifying moiré impacted images.

METHODOLOGY

The implementation of the project was done using PyTorch with Torch Vision due to its relative ease of use and clear documentation. A multitude of research exists already that has made use of PyTorch making it a suitable choice.

A video feed on a virtual production film set is unlikely to be in a resolution less than 4k. For that reason, a dataset of 4k images seems like a suitable choice for this research. As an image contaminated with spatial aliasing is downscaled there is a potential risk that the low frequency aliasing will be lost in the image. It is important that the input data the CNN is being trained on retains this aliasing. A potential solution would be to sample smaller sections of an image containing moiré patterns in order to speed up training and allow more training epochs to take place. This solution however will not work without a custom labelled dataset due to the risk of sampling a section of an image where aliasing isn't present and training the CNN with incorrectly labelled data. As sampling and considerable downscaling was not an option it was decided to convert the images from RGB to grayscale in order to reduce the channels from 3 to 1 as aliasing still appears to be present in this case.

A powerful compute server was used for training with such high-resolution images however development was carried out on a standard workstation not capable of such high video memory. For this reason, the solution automatically detects the presence of CUDA compatible GPUs with CPU fallback and seeds across PyTorch, CUDA and Python's random module for reproducibility. cudnn.benchmark = True optimises kernel selection for GPU performance.

The solution begins by applying some initial transforms to each image ensuring it is of tensor format, a fixed image size and randomly flipped for data augmentation. The fixed image size that was selected is half that of the original image size in the dataset. 2016 pixels wide by 1512 pixels in height. This size appears sufficiently downscaled to be processable while still retaining the spatial aliasing artefacts in the original data.

For simplicity TrivialAugmentWide was used for data augmentation as there were no augmentations that seemed like they would have a particularly adverse effect on the integrity of the data. This augmentation strategy maintains consistency between the training and test transformations. The images are also converted to grayscale in this stage to reduce the channels to 1. Data augmentation improves generalisation for training and converting the images to grayscale simplifies the input data without sacrificing discriminative power.

The training and testing datasets were both set up using DataLoader and configured with a batch size of 8 to make use of as much video memory as possible. Zero worker threads were used for data loading to allow for maximum compatibility when working in restricted environments such as the JupyterLab environment that exists on the compute server. The DataLoader efficiently handles mini-batching and shuffling of the data for stochastic optimisation. This is essential for stochastic gradient descent (SGD). The optimiser used in this instance was Adam (Kingma & Ba, 2015) as it required less hyperparameter optimisation, converged faster and there was only limited training time available.

The system uses a custom model defined in the ImageClassifier() module. The model is composed of multiple convolutional blocks followed by a sequence of fully connected layers.

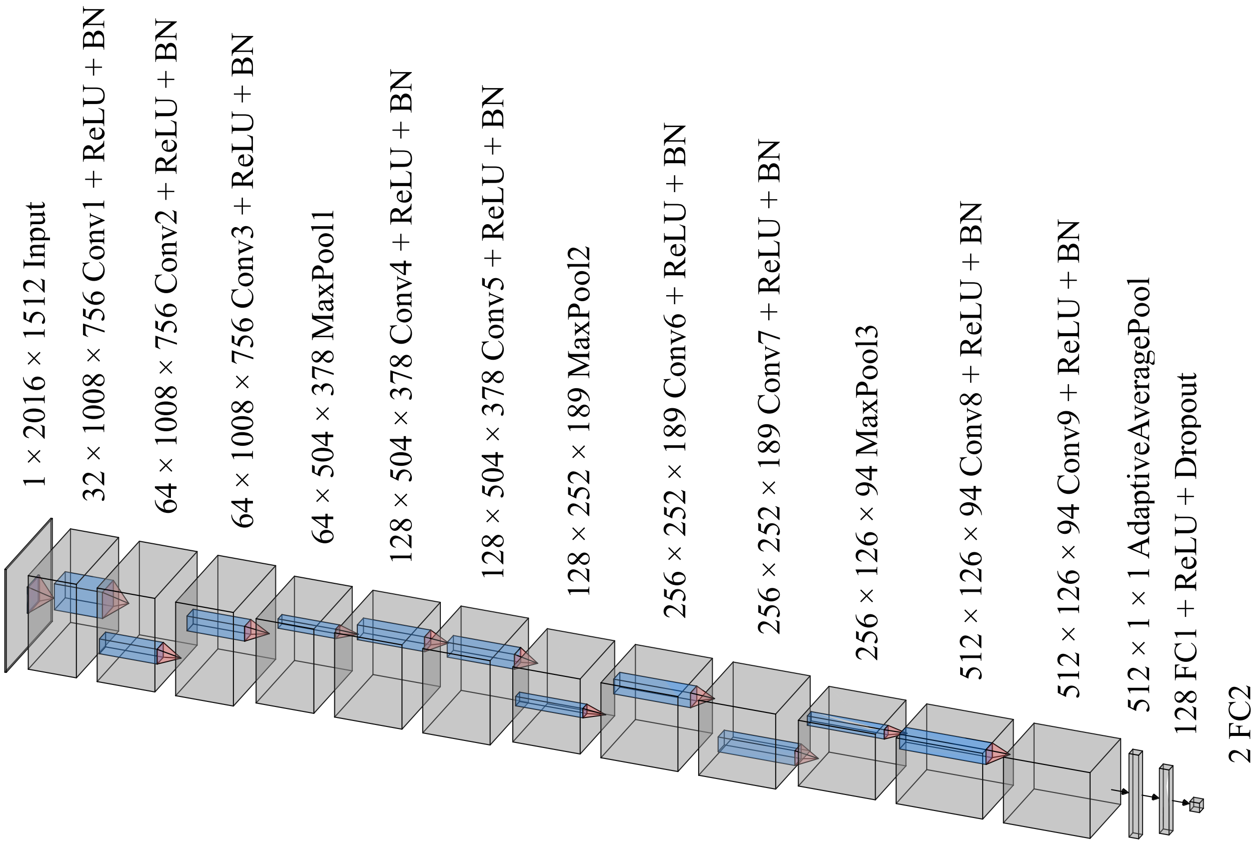

Figure 5. CNN Model Architecture Diagram

The input transformed tensor takes a down scaled image from the dataset with a single grayscale channel. The first convolution layer has a large kernel size to capture broad content early while downsampling to 1008 × 756 to reduce compute costs. This is followed by batch normalisation (BR) and ReLU to improve convergence and add non-linearity (Ioffe & Szegedy, 2015) (Nair & Hinton, 2010). This initial convolution layer is a compromise between compute efficiency and early information capture.

Multiple feature blocks then follow the pattern of two 3 × 3 convolutional layers, an initial convolutional layer followed by a deeper convolutional layer. ReLU and BR are then implemented after for non-linearity. A max pooling layer then halves the resolution. The output of the first block is 64 × 504 × 378, the 64 channels is double the capacity of the initial layer allowing the model to learn more refined edges. Pooling then reduces the size while preserving key features. The second block 128 × 252 × 189 learns more abstract features such as textures and corners with 128 channels. The third block outputs 256 × 126 × 94 to learn even more complex patterns and part structures (Zeiler & Fergus, 2014).

The final deep feature block with no pooling contains 512 feature channels to capture high-level semantic features. Any further pooling beyond 126 × 94 risks information loss. The adaptive average pooling layer then collapses each 2D feature map into a single value with an output shape of 512 × 1 × 1. This layer generalises by summarising the entire feature map.

The first fully connected layer flattens 512 × 1 × 1 into a 512-dimensional vector before being passed into the first fully connected linear layer where it is reduced to 128 units to limit overfitting. The model learns high-level combinations of features before randomly dropping out 30% of the activations to reduce overfitting.

The second fully connected layer produces raw logits for two-class classification. These logits represent the networks confidence in each class before applying an activation function. The model outputs raw logits that are passed through the Softmax function when inferencing the model.

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Softmax function

The model's architecture allows for multiple GPU support using nn.DataParallel if multiple GPUs are available. This is particularly important due to the large dataset and availability of two Nvidia H100 cards. The custom design of the image classifier model is tailored towards high resolution grayscale images making training more effective on the powerful server.

The training process used CrossEntropyLoss as the loss function, otherwise known as log loss.

$$L = -\sum_{i=1}^{C} y_i \log(\hat{p}_i)$$

Cross entropy loss function.

In this loss function, L is the loss, y_i is the ground truth label for class i and $\hat{p}_i$ is the predicted probability for class i (Goodfellow, et al., 2016).

The Adam optimiser is an improved version of SGD. Typical SGD uses a fixed learning rate that must be manually tuned (Forsyth & Ponce, 2002), Adam however is adaptive per parameter. It utilises built-in moving averages and converges faster. (Kingma & Ba, 2015)

$$\theta = \theta - \eta \cdot \nabla_{\theta L(\theta)}$$

Basic SGD optimiser.

$$\theta = \theta - \eta \cdot \frac{m_t}{\sqrt{v_t} + \epsilon}$$

Improved Adam optimiser.

Each epoch loop consists of a training step and testing step using the trainbatch() and testbatch() functions. It also records the training and testing loss and accuracy in order to monitor the efficacy of the model. The monitoring of these metrics helps with the diagnosis of overfitting and underfitting during training cycles.

Once the training of the model has been completed the weighted model is saved and automatically versioned to compare against previous models. Custom image inference can then be carried out to test the efficiency of the trained model. Images are first pre-processed by resizing and converting to grayscale before being transformed into an input tensor with batch dimension. The Softmax probabilities and predicted class are then outputted.

$$x \in R^{1\times H\times W}$$

$$x_1 = BN_1(ReLU(W_1 * x + b_1))$$

$$x^2 = BN^2\big(ReLU(W^2 * x^1 + b^2)\big)$$

$$x^3 = BN^3\big(ReLU(W^3 * x^2 + b^3)\big)$$

$$x^4 = MaxPool(x^3)$$

$$x^5 = BN^4\big(ReLU(W^4 * x^4 + b^4)\big)$$

$$x^6 = BN^5\big(ReLU(W^5 * x^5 + b^5)\big)$$

$$x^7 = MaxPool(x^6)$$

$$x^8 = BN^6\big(ReLU(W^6 * x^7 + b^6)\big)$$

$$x^9 = BN^7\big(ReLU(W^7 * x^8 + b^7)\big)$$

$$x^{10} = MaxPool(x^9)$$

$$x^{11} = BN^8\big(ReLU(W^8 * x^{10} + b^8)\big)$$

$$x^{12} = BN^9\big(ReLU(W^9 * x^{11} + b^9)\big)$$

$$x^{13} = AdaptiveAvgPool(x^{12})$$

$$x^{14} = Flatten(x^{13})$$

$$x^{15} = Dropout(x^{14}, p = 0.3)$$

$$x^{16} = ReLU\big(W_{fc1} \cdot x^{15} + b_{fc1}\big)$$

$$x^{17} = Dropout(x^{16}, p = 0.3)$$

$$\hat{y} = W_{fc2} \cdot x^{17} + b_{fc2}$$

CNN forward pass

DATASET

The training dataset contains 10,000 images of moiré present and ground truths in equal split. The "Ultra High-definition Demoiréing Dataset" was created by the authors of the paper "Towards Efficient and Scale-Robust Ultra-High-Definition Image Demoiréing" (Yu, et al., 2022) making it perfect for the training of this machine learning model. The creators of the dataset used a number of control methods to ensure a diverse collection of moiré patterns was produced. This includes the use of a gimbal to flexibly adjust the camera as well as utilising a variety of screens and cameras. It would have been optimal to produce a custom dataset of virtual production moiré images however this data was not available at the time of the project and creation of the data would have fallen outside the scope of the project.

The acquisition of the dataset was done by downloading it from the Kaggle source (Kaggle, 2022). As stated in the report by Yu, et al. this is the first high resolution demoiréing dataset of its kind making it an obvious choice for a project with limited resources available. The images have all been captured in high resolutions ranging from 4k to 8k and then normalised to easily parse into an input tensor. This manual normalisation ensures that the aliasing artefacts are still present in all training data.

Potential biases exist in the UHDM training data due to the way in which it was captured could make it inappropriate for the context of aliasing in virtual production. Device bias poses a real risk due to the capture devices all being consumer smartphones and not professional virtual production cameras. An optimal dataset would make use of professional, full-frame cameras built for virtual production such as the Canon C400 or Sony Venice. The data may not accurately reflect the behaviour of a cinecamera's optics, sensor behaviour, or the rolling shutter artefacts often found in virtual production.

The data contains varied scenes and scenarios such as landscapes and documents. While this variation is good for general demoiréing use cases to prevent overfitting and encourage generalisation, it does not accurately reflect the likely real-world data that the model's implementation will come into contact with. The UHDM dataset lacks LED volume content that is commonly seen in virtual production meaning the moiré artefacts and spatial aliasing is not LED specific. This type of interference pattern is spatially structured, screen dependent and dynamic.

The way moiré is introduced in this dataset is not realistic to the angles and shooting geometries seen in virtual production. The lighting is also not representative of that seen in a virtual production environment. The images are all uniformly 4k, this is advantageous for easy training as previously mentioned however in a virtual production setting non-standard aspect ratios may be used. There is a risk of the model overfitting to 16:9 full frame samples. Ideally a variety of resolutions and formats would be adopted to accurately reflect the multitude of camera sensors that could be used.

It is unknown what postprocessing functions have been performed on the images in the dataset. Often consumer phone cameras introduce artificial mosaicking or sharpening algorithms that could have a negative impact on the integrity of the dataset for virtual production.

Ideally a dataset for this use case would be custom made and consist of a varied collection of image samples representative of virtual production work. Footage shot across various LED panel manufacturers with a variation of resolutions, pixel pitch, distance from wall, varied lighting setups would improve the quality of the dataset for virtual production. As well as varied wall manufacturers and lighting, a wide range of camera bodies, sensor sizes, aspect ratios and lenses would improve the dataset for the proposed use case. Despite the concerns with the dataset for virtual production, it is a good dataset and with limited resources available it is sufficient in aiding research on this topic. (Wen, et al., 2021)

RESULTS AND ANALYSIS

The most accurate model was produced on the 10th training iteration after approximately 86 hours of training. Further hyperparameter optimisation could likely be carried out but due to the resources available this model is the current most suitable for the given use case.

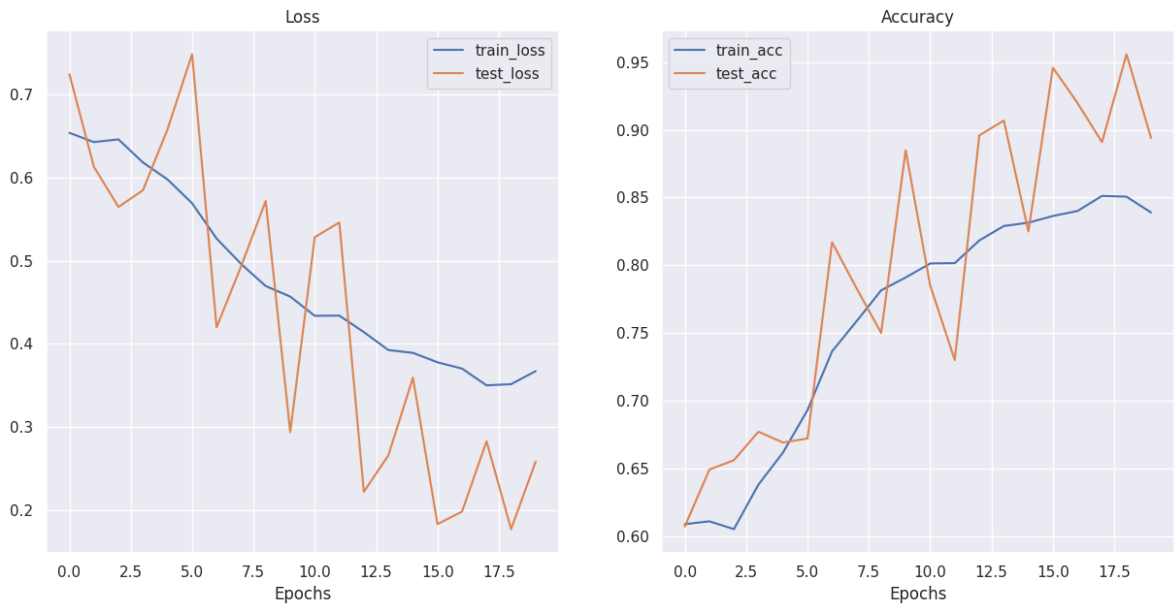

Figure 6. Loss vs accuracy graphs generated during model training

When testing the model on a dataset of 1000 images containing moiré and clean in even split the model achieved the following results.

	Predicted Moiré	Predicted No Moiré
Contains Moiré	431	37
No Moiré	37	463

Figure 7. Confusion matrix of the trained model

The key functions to analyse the efficiency of the model of predicting the given data include:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{431 + 463}{431 + 463 + 37 + 37} = \frac{894}{968} \approx 0.924$$

$$\text{Precision (Moiré)} = \frac{TP}{TP + FP} = \frac{431}{431 + 37} = \frac{431}{468} \approx 0.921$$

$$\text{Precision (No Moiré)} = \frac{TN}{TN + FN} = \frac{463}{463 + 37} = \frac{463}{500} \approx 0.926$$

The precision of both moiré and no moiré are very close suggesting the model is performing well in both classes. With an accuracy of 92.4% the model appears well tuned for this binary classification task. With 7.6% inaccuracy there is room for improvement through finer hyperparameter optimisation.

CONCLUSION

In conclusion, the training of a deep learning model using a convolutional neural network is a novel approach to the detection of moiré artefacts in an image. Moiré, a form of spatial aliasing is a prominent problem in the virtual production industry, contaminating cinematic shots and costing production companies significantly in post-production visual effects work. Inferencing a trained deep learning model with sampled images from a camera feed offers the ability to detect aliasing early in the filming process alerting on-set crew to its presence.

When training deep learning CNNs for the detection of low frequency aliases of high frequency details in poorly sampled images it is important to maintain a large resolution. During the project training in its initial form, the training dataset was down-sampled significantly more than it is currently which resulted in overfitting due to the moiré artefacts being lost and the model being trained on falsely labelled data.

It would be appropriate in future research to develop a more suitable dataset that accurately portrays the aliasing artefacts seen in virtual production as well as limiting the dataset biases present in the UHDM dataset used in this research project. A larger collection of LED panels, with a variety of lighting conditions representative of that seen on an actual virtual production set would prevent the model overfitting; as would a wider variety of professional camera equipment including bodies and lenses.

AI ACKNOWLEDGEMENT

It is important to acknowledge the use of large language models (LLMs) in this project. LLMs were used to amplify my personal development and knowledge building. Prompts were used to understand sections of code and provide explanations along with the official documentation and existing research in this field. A log of LLM prompts can be found alongside the project submission.

REFERENCES

Abraham, E., 2018. *Moiré Pattern Detection using Wavelet Decomposition and Convolutional Neural Network.* Bangalore, IEEE.

Brompton Technology, 2025. *Let's Talk About Moiré.* [Online]
Available at: https://www.bromptontech.com/lets-talk-about-moire/
[Accessed 2 May 2025].

Crowley, P., 2005. *An intuitive guide to wavelets for economists,* Helsinki: Bank of Finland Research Discussion Papers.

Data Physics, 2016. *Dynamic Signal Analysis Review: Part 3 – Aliasing Frequency.* [Online]
Available at: https://dataphysics.com/blog/dynamic-signal-analysis/dynamic-signal-analysis-review-part-3-aliasing/
[Accessed 2 May 2025].

Forsyth, D. & Ponce, J., 2002. *Computer Vision: A Modern Approach.* New Jersey: Prentice Hall Professional Technical Reference.

Goodfellow, I., Bengio, Y. & Courville, A., 2016. *Deep Learning.* Massachusetts: The MIT Press.

Ioffe, S. & Szegedy, C., 2015. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.* Lille, ICML.

Kadner, N., 2024. *Scalable Virtual Production—Taking a "Just Right" Approach.* [Online]
Available at: https://blog.frame.io/2024/01/15/scalable-virtual-production-small-medium/
[Accessed 3 May 2025].

Kaggle, 2022. *Ultra High-definition Demoiréing Dataset.* [Online]
Available at: https://www.kaggle.com/datasets/soumikrakshit/uhdm-dataset
[Accessed 4 May 2025].

Kingma, D. & Ba, J., 2015. *Adam: A Method for Stochastic Optimization.* San Diego, ICLR.

Mahya Hazavei, S. & Reza Shahdoosti, H., 2017. *A New Method for Removing the Moiré Pattern from Images,* Hamedan: Arxiv.

Maksim, 2005. *File:Moire pattern of bricks small.jpg.* [Online]
Available at: https://commons.wikimedia.org/wiki/File:Moire_pattern_of_bricks_small.jpg
[Accessed 2 May 2025].

Nair, V. & Hinton, G., 2010. *Rectified Linear Units Improve Restricted Boltzmann Machines.* Haifa, ICML.

Netflix, 2025. *Common Virtual Production Challenges & Potential Solutions.* [Online]
Available at: https://partnerhelp.netflixstudios.com/hc/en-us/articles/1500002086641-Common-Virtual-Production-Challenges-Potential-Solutions
[Accessed 2 May 2025].

Patney, A. & Lefohn, A., 2018. *Detecting Aliasing Artifacts in Image Sequences.* Vancouver, Association for Computing Machinery.

Ribeiro, A. H. & Schön, T. B., 2021. *How Convolutional Neural Networks Deal with Aliasing.* Toronto, IEEE.

Rigard, 2025. *How to Fix the Moire Effect on LED Screens.* [Online]
Available at: https://rigardled.com/how-to-fix-the-moire-effect-on-led-screens/
[Accessed 2 May 2025].

Shannon, C., 1998. Communication in the Presence of Noise. *IEEE,* 86(2), pp. 447-457.

Steve Marschner, P. S., 2016. *Fundamentals of Computer Graphics.* 4th ed. Boca Raton: CRC Press.

Tomas Akenine-Möller, E. H. N. H., 2008. *Real-Time Rendering.* 3rd ed. Wellesley: A K Peters Ltd..

Vetterli, M., Kovačević, J. & Goyal, V., 2014. *Foundations of Signal Processing,* Cambridge: Cambridge University Press.

Wen, Z., Wang, H., Gong, Y. & Wang, J., 2021. Denoising convolutional neural network inspired via multi-layer convolutional sparse coding. *Journal of Electronic Imaging,* 30(2).

Yu, X. et al., 2022. *Towards Efficient and Scale-Robust Ultra-High-Definition Image Demoireing.* Hong Kong: The University of Hong Kong.

Zeiler, M. & Fergus, R., 2014. *Visualizing and Understanding Convolutional Networks.* Zurich, Springer.

Zhang, R., 2019. *Making Convolutional Networks Shift-Invariant Again.* California, ICML.

Zhuge, J., 2024. *Dynamic Signal Analysis Basics.* [Online]
Available at: https://www.crystalinstruments.com/dynamic-signal-analysis-basics
[Accessed 2 May 2025].