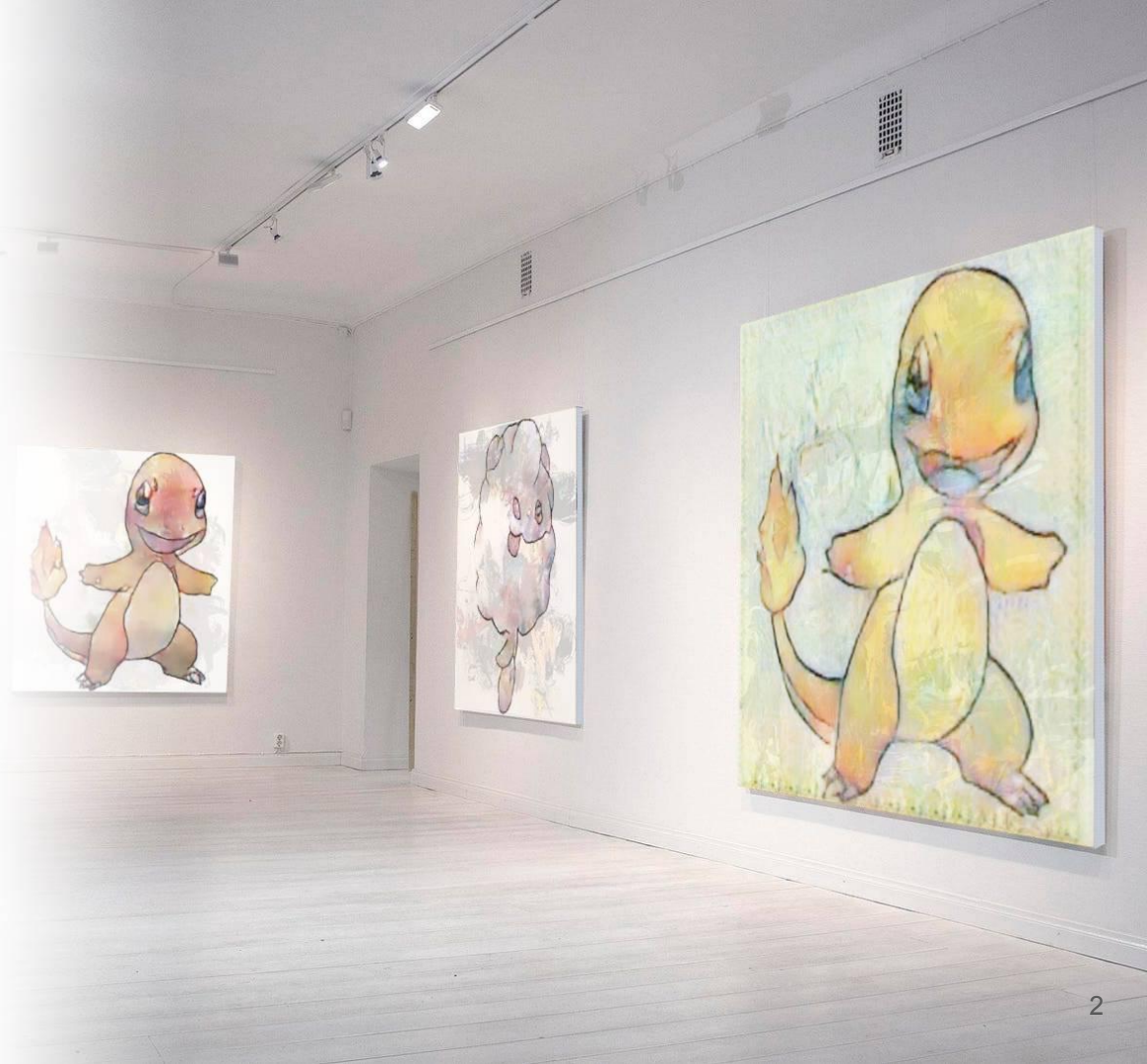
NFTGan - Draw Masterful Art Pieces

Presented by Group 21

Mu Aohua - A0121924M
Lee Joon Hui Jeremy - A0048174A

What is NFTGan?

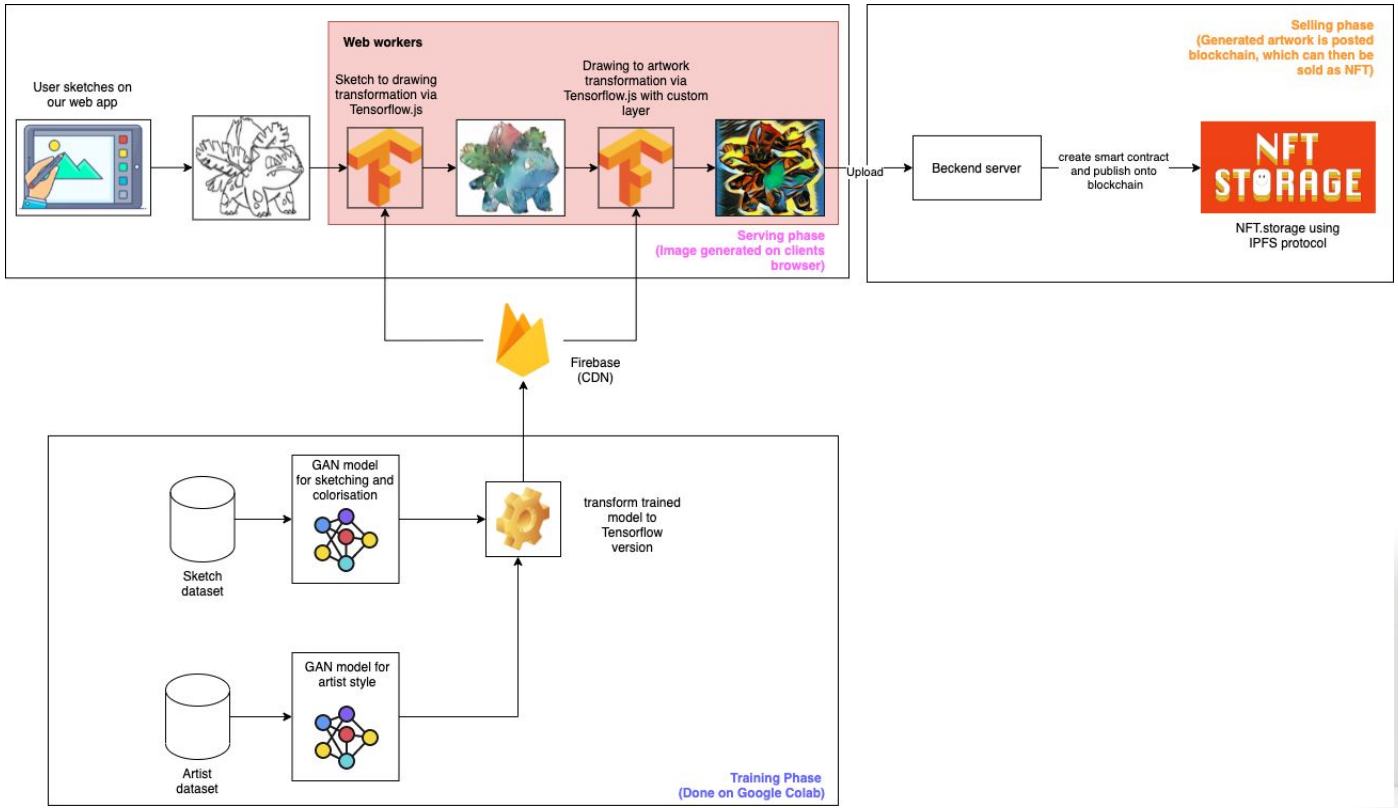
It is a web application that allows anyone to create masterful artwork with just a sketch

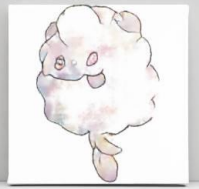


Project Objective

1. Understand how GAN works
2. Productionise the models into practical application using TensorflowJS
3. Improve existing GAN models with more suitable datasets or the model’s structures

Final System Design





Training Phase

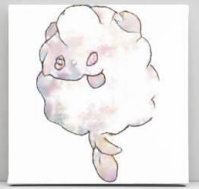
We use 2 GAN models to translate a sketch into an artwork

Pix2Pix

Convert sketch into coloured drawing

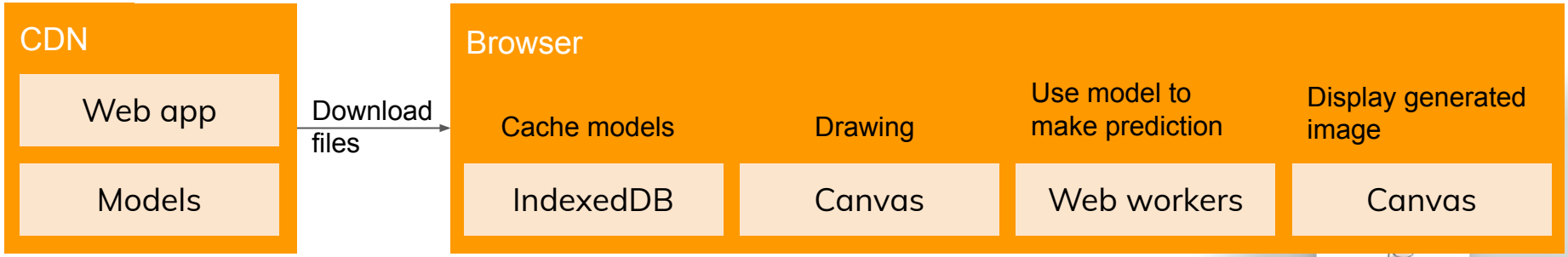
CycleGAN

Convert drawing into art pieces

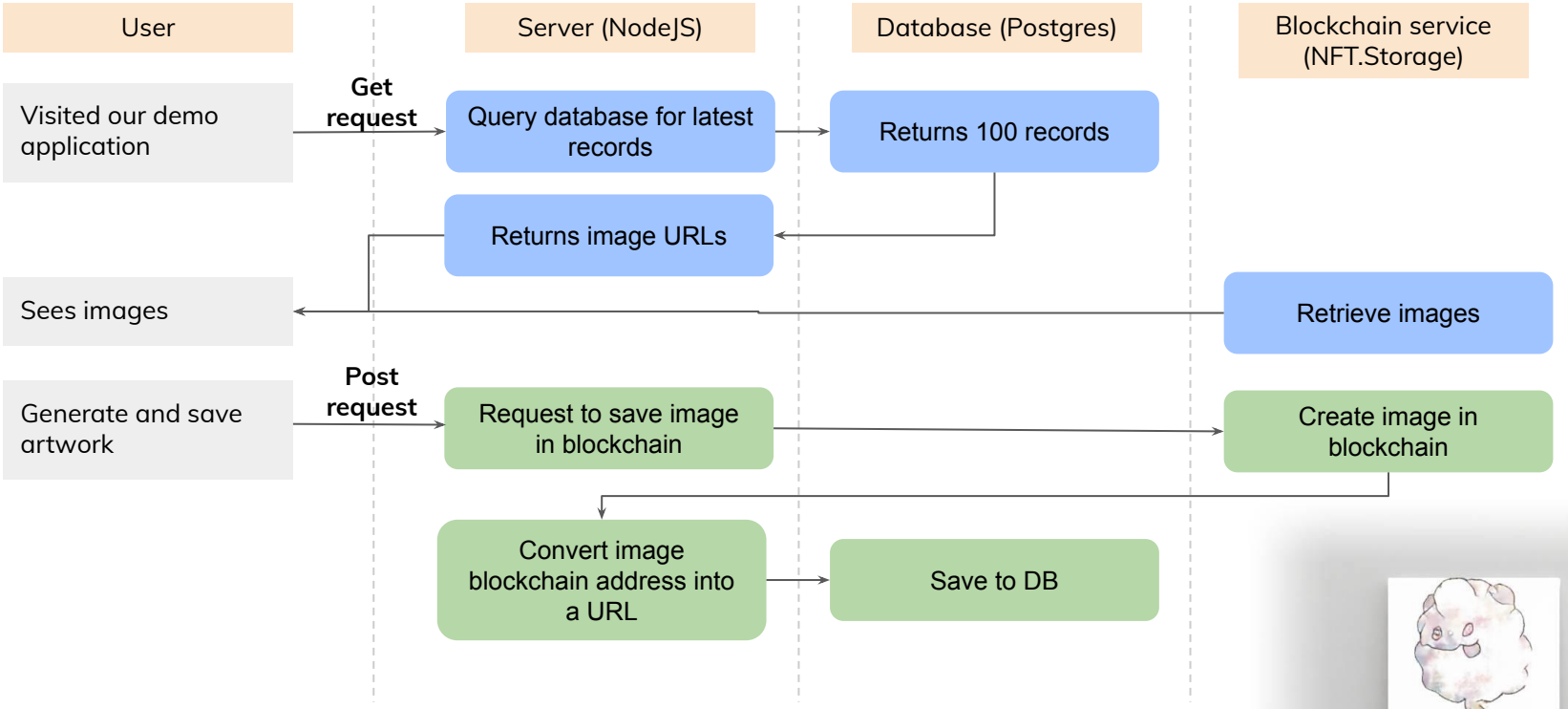


Serving Phase

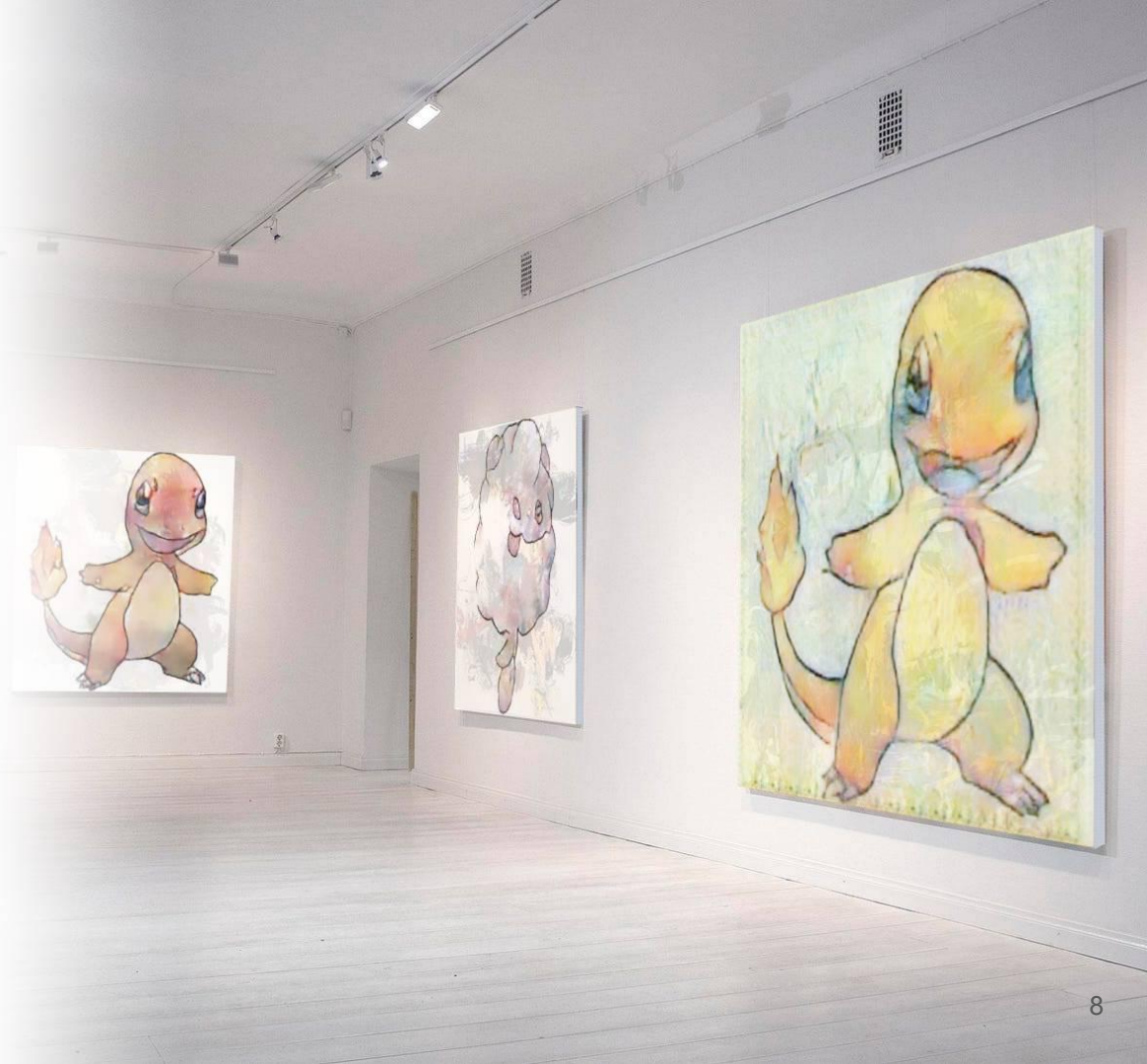
1. Our sketching board and models are served as **static files** and **fully running in the browser**.
2. Utilise **web workers** to handle computational complex tasks(e.g. Model prediction).
3. **Custom instance normalization layer** for UNet
4. Models are cached in browser **indexedDB**



Selling Phase



Demo



Demo

Demo

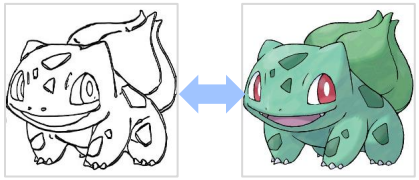
Colorisation

Final art piece

Try it yourself at: https://draw-it-yourself.web.app/

Dataset Used and Feature Engineering

Pix2Pix

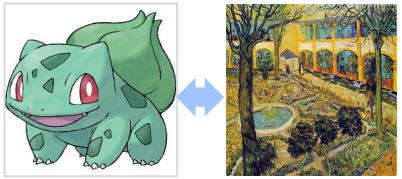


`sketch2drawing`

CycleGAN

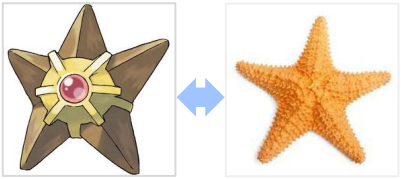
`vangogh2photo`

Serves as our baseline



`pokemon2vangogh`

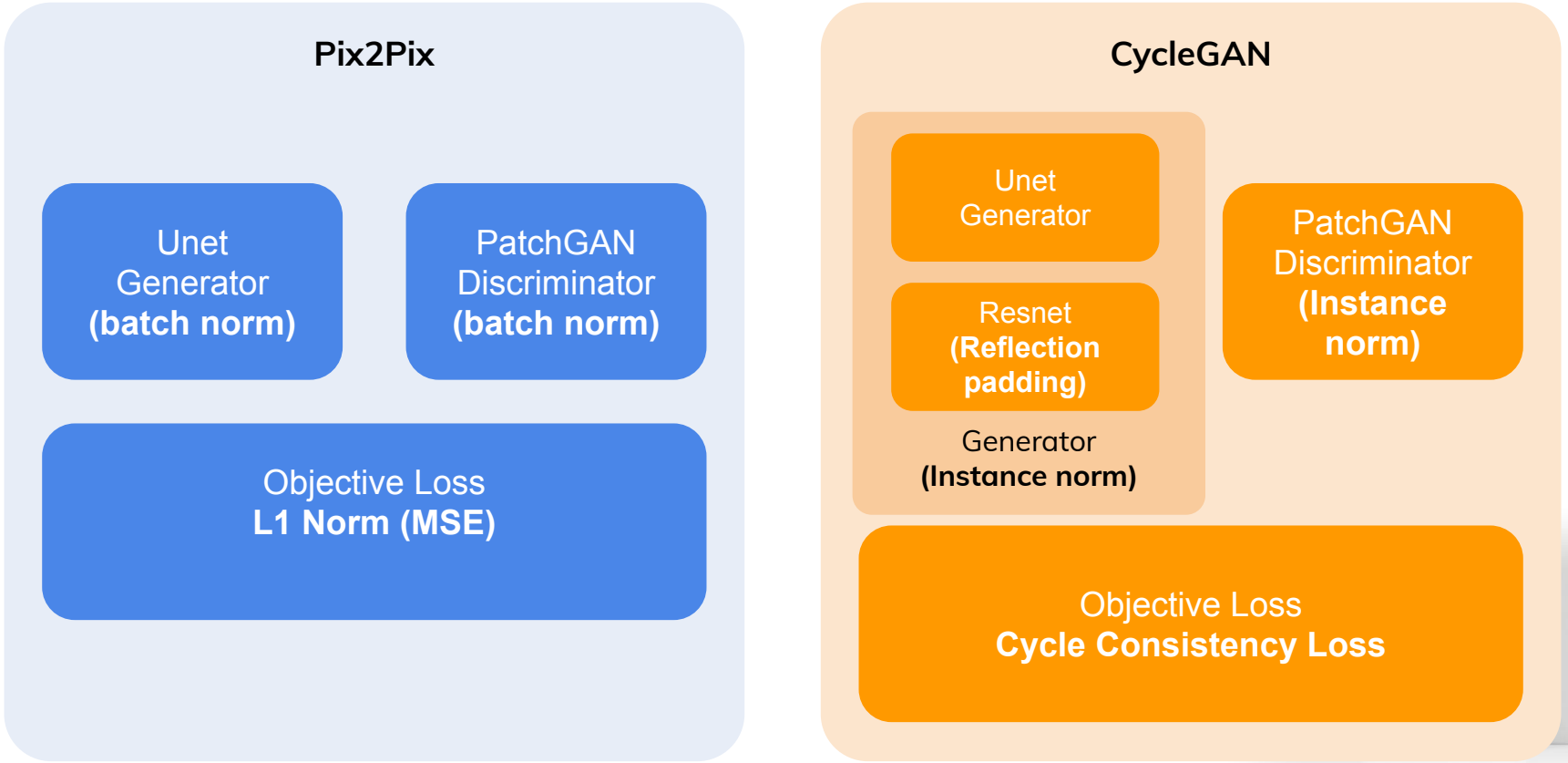
Helps to generalise results better



`pokemon2animals`

Using same image domain to improve models

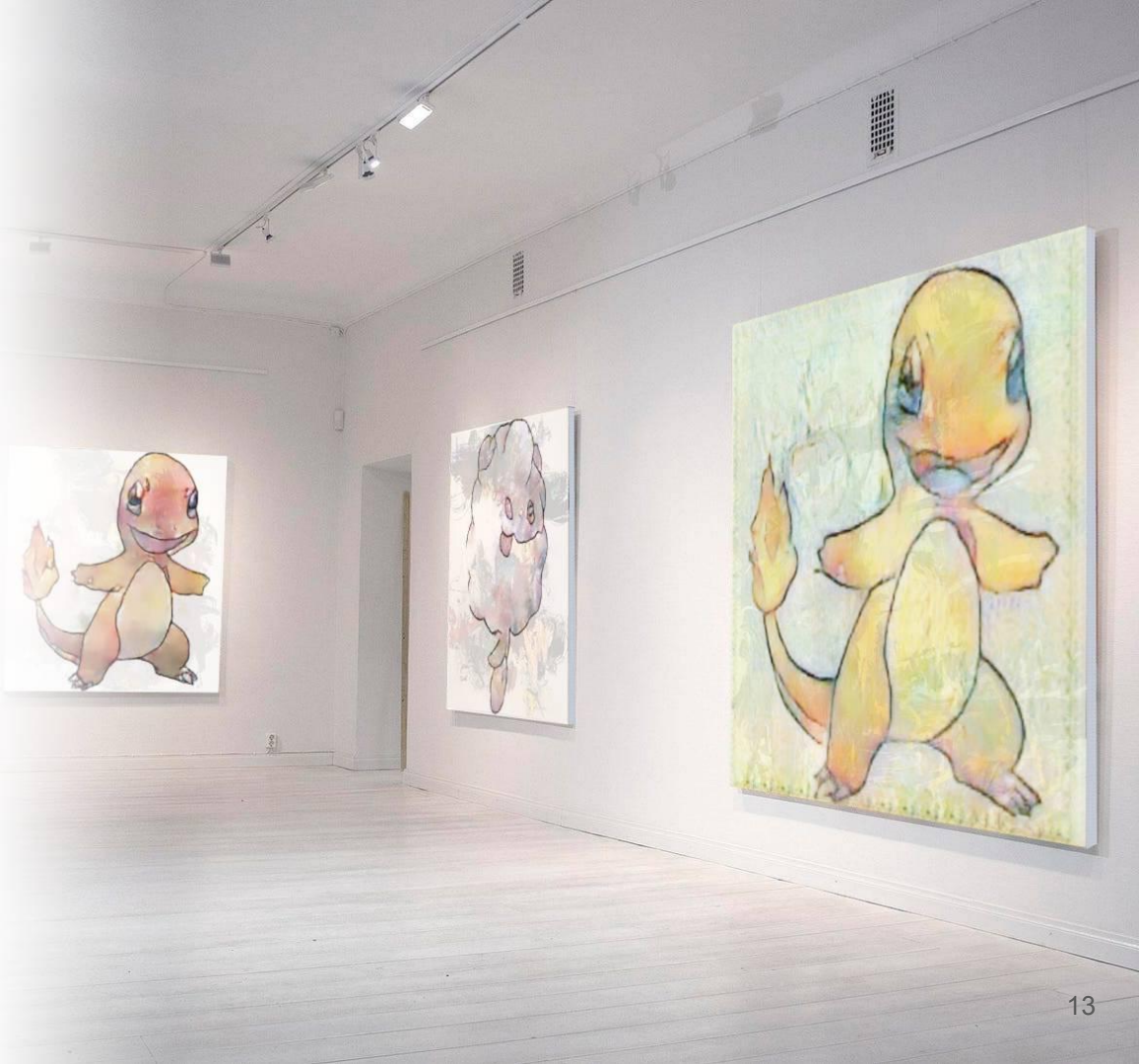
Models Used


Pix2Pix
Unet Generator (batch norm)
PatchGAN Discriminator (batch norm)
Objective Loss L1 Norm (MSE)
CycleGAN
Unet Generator
Resnet (Reflection padding)
Generator (Instance norm)
PatchGAN Discriminator (Instance norm)
Objective Loss Cycle Consistency Loss

Evaluation and Performance

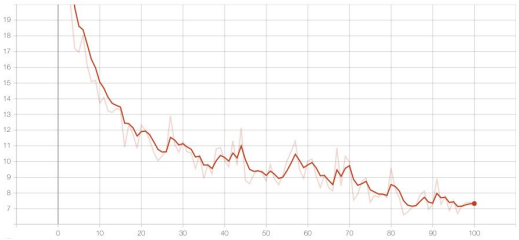
1. Loss metrics to monitor training performance

2. Polling of participants to vote for the model that produces best results



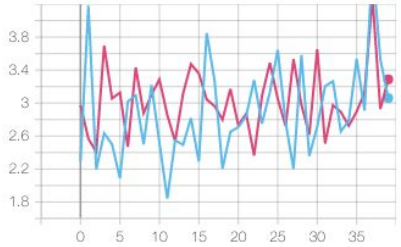
Loss Metrics

Pix2Pix



The total loss trends downwards, which indicates that the model is generalisation **well**

CycleGAN



The cycle consistency loss did not generalise well for both unet and resnet models, cannot determine the best model for the app, have to rely on polls to decide

Poll Results

Pix2Pix

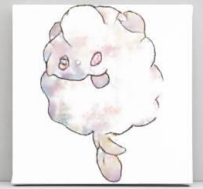
Image			
Type of photos	Actual photo	Generated by our model	Actual photo
Expected category	Photo of a City	Drawing of a pokémon	Painting of a night sky
Responses	97.4% correct	**100% correct**	100% correct

100% of our participants are able to classify our generated image correctly

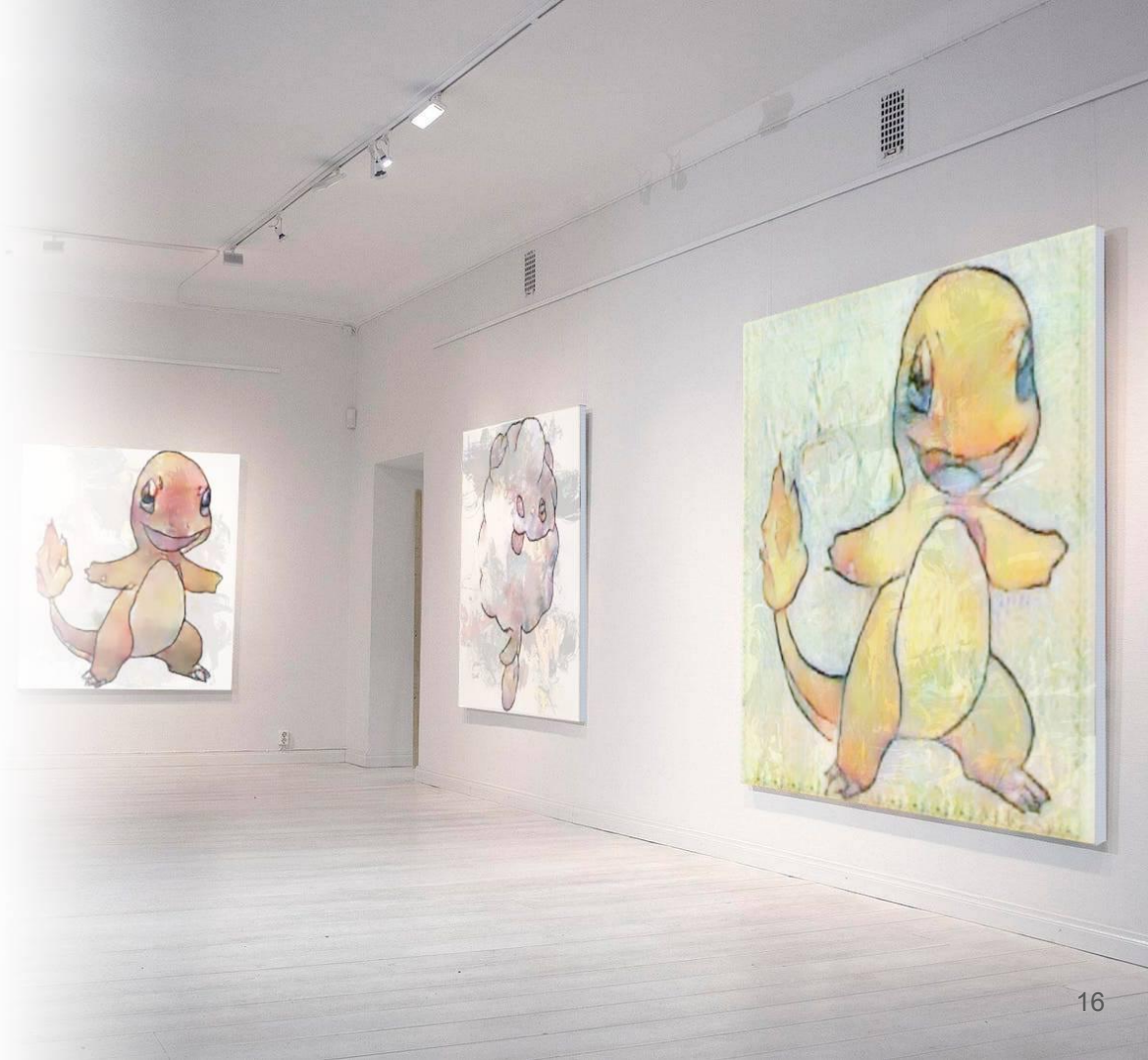
CycleGAN

	Predicted Image A (pokemon2animals dataset)	Predicted Image B (pokemon2vangogh dataset)	Predicted Image C (vangogh2photo dataset)
Average score for CycleGan	4.538461538	5.692307692	**6.820512821**
S.D. for CycleGan	2.382459519	2.318732028	2.186944454

The best performing model is the one trained using `vangogh2photo` with an average score of 68%!

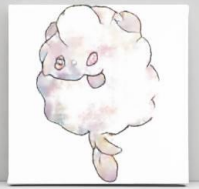


Learnings



Serving Phase - Learning 1/3

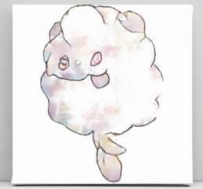
Displaying of generated image on the browser UI takes time. This is because the process of converting the model's output array into tensors locks up most of the UI resources.



Serving Phase - Learning 2/3

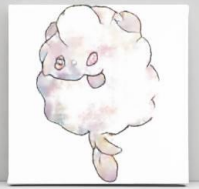
We hope to achieve **real time rendering** of our generated image while **balancing model execution frequency**, which can be resource intensive.

We propose to use a better browser debouncing technique that could balance the two constraints.



Serving Phase - Learning 3/3

TensorFlow.js Layers currently only supports Keras models using standard Keras constructs and not those with unsupported ops or layers such as custom layers, Lambda layers, custom losses, or custom metrics. This means that we cannot automatically import our GAN that uses our custom instance normalisation layer.



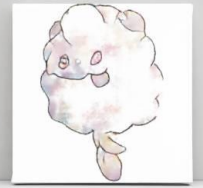
Selling Phase - Learnings

Existing AI technology is relatively straightforward to embed into existing software application thanks to tools like Tensorflow and Keras.

Training Phase - Learning 1/2

GAN performance is better evaluated by human instead of algorithm due to the subjectivity of the output

100% of participants correctly classified our model generated image as pokemon

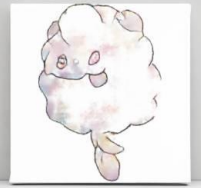


Training Phase - Learning 2/2

Datasets plays an equally important role to determine the performance of the model

vangogh2photo

Model trained by baseline data performs better than handcrafted data



Conclusion

We set out to learn the inner workings of the Generative Adversarial Network by creating an drawing MVP for this project.

What we acquired is a deeper understanding of deep neural networks and how they can fuse together to create wonders.

We are also grateful to learn the concepts in this course because it helps us to understand the concepts and terminologies much easier.