Voice to Image Generation

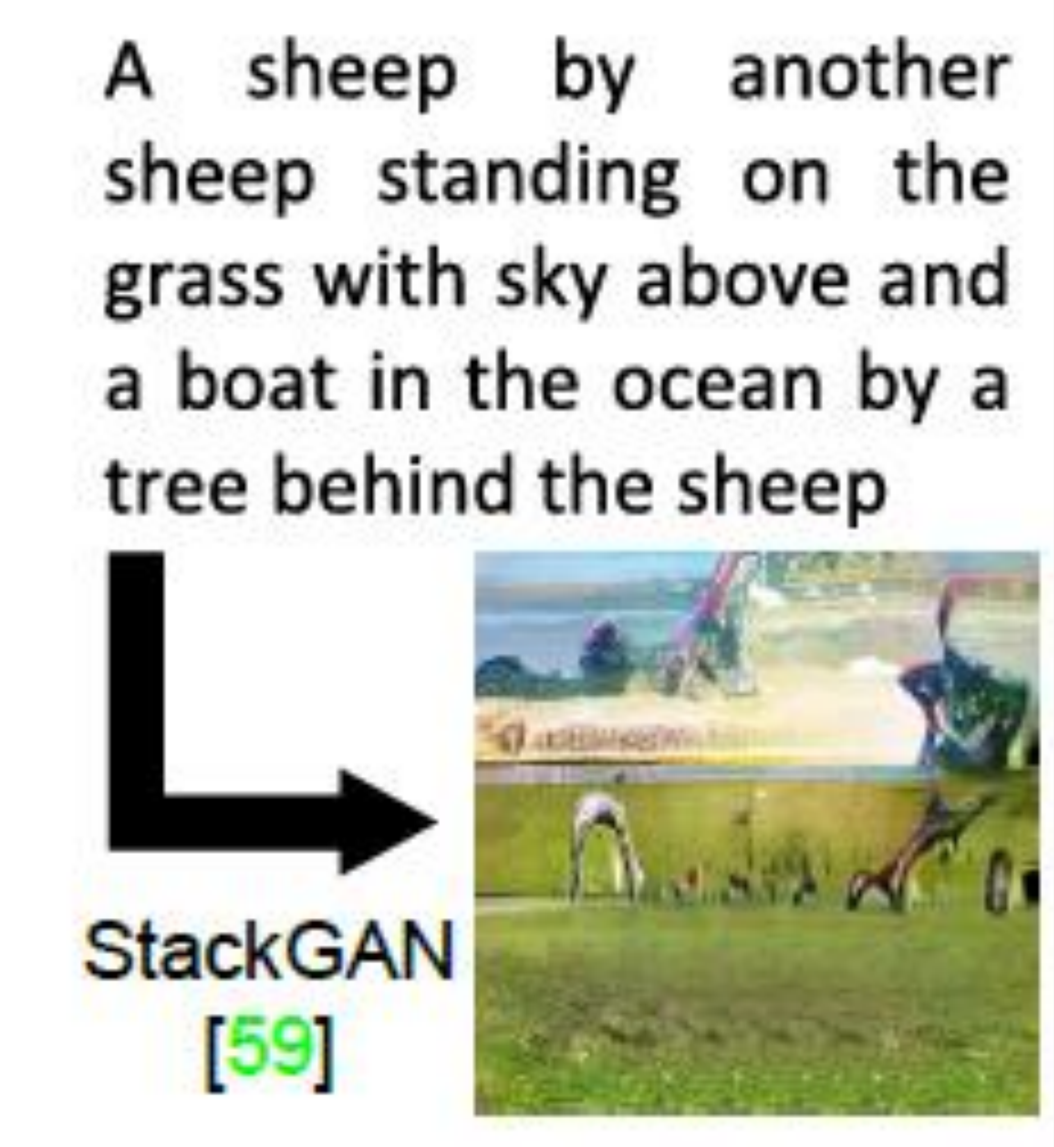
David Ekvall; Sabeen Nawaz; Qingyan Ben

Overview

- Idea
- Approach – Three separate parts
- Speech to text
    - Results
    - Discussion
- Text to image
    - Results
    - Discussion
- Conclusion
- Learning outcomes

Idea - Automatic image generation

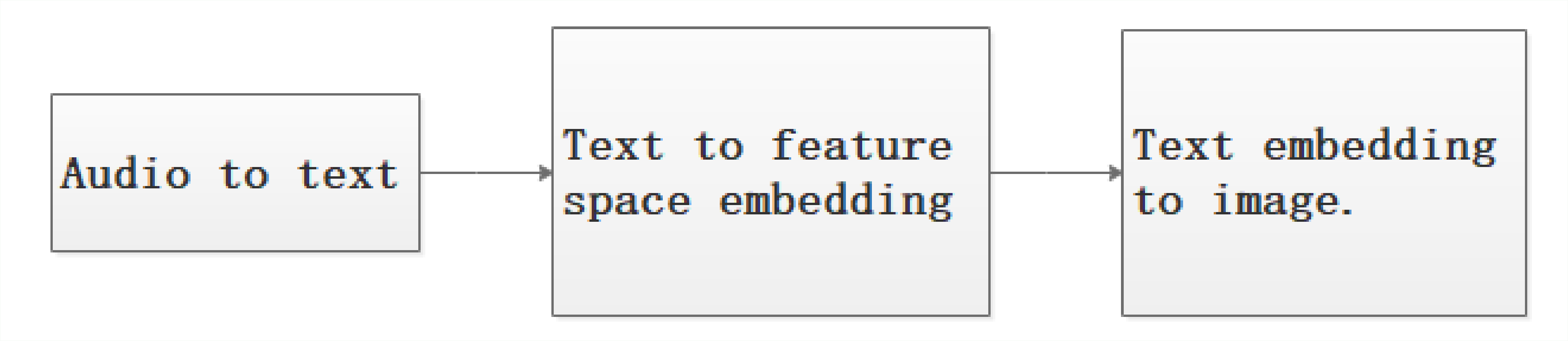
- Generate an image from spoken sentence
- Could be used for artistic purposes etc.
- There exists methods to generate images from natural language descriptions.



A sheep by another sheep standing on the grass with sky above and a boat in the ocean by a tree behind the sheep

Approach - Model structure

- Three parts
  - K-layer neural network
  - Word2vec
  - Conditional GAN



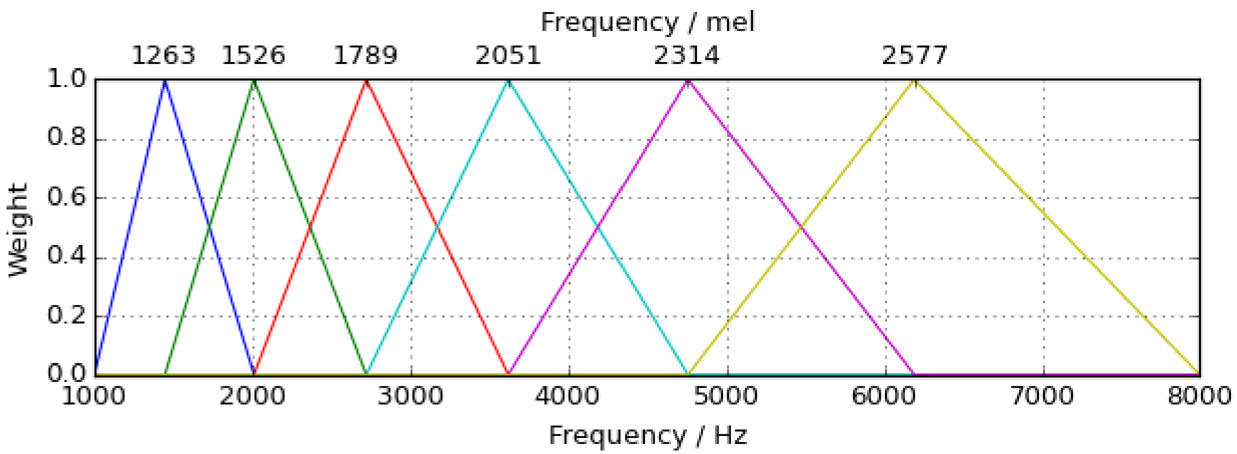
Audio to text

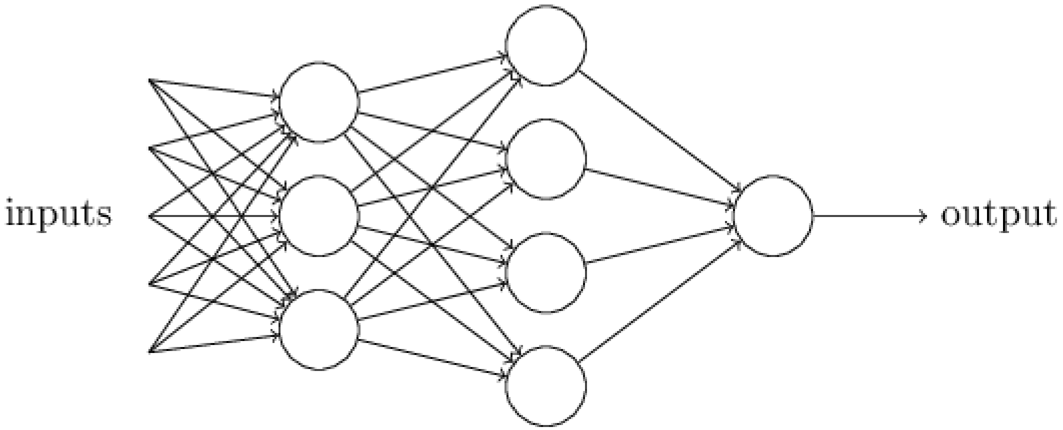
Text to feature space embedding

Text embedding to image.

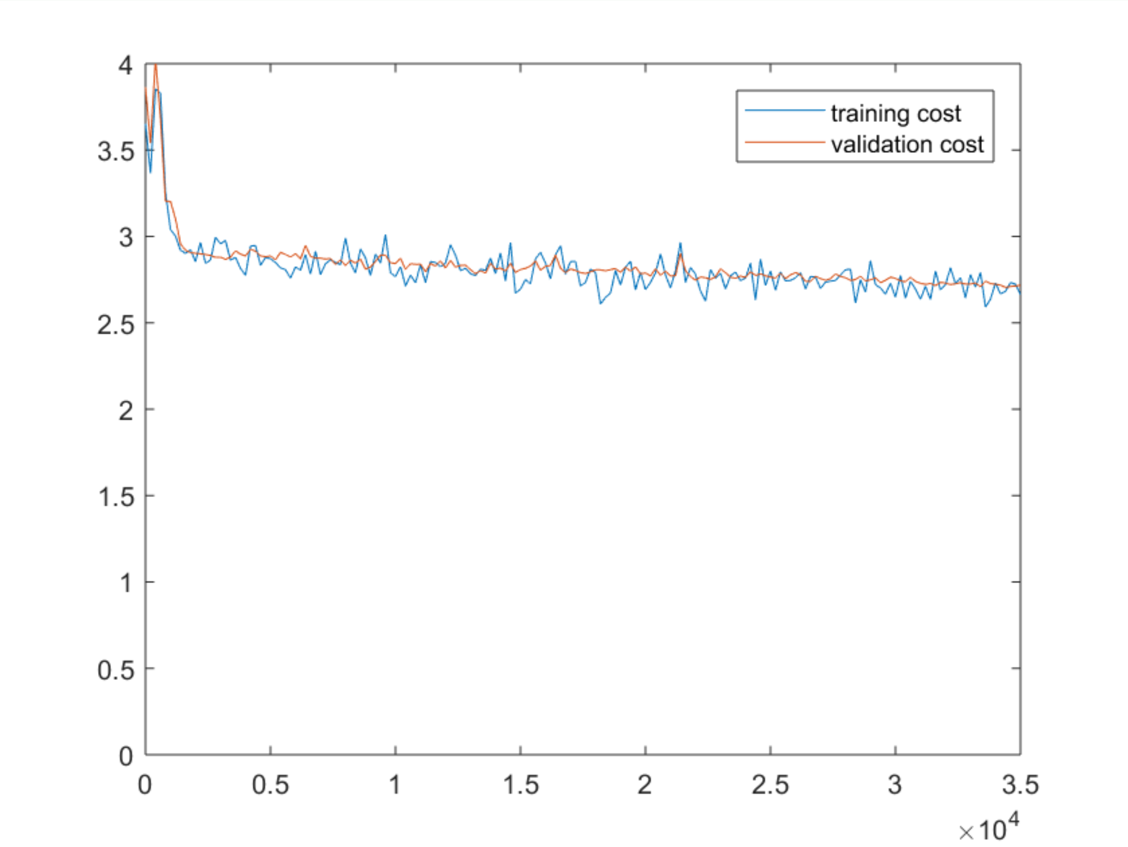
Approach - Speech to text

- Label spoken words
- Thirty words classes
- Preprocessed to Mel's frequency cepstral coefficient vectors
- Trained with cyclic learning rates and batch normalization
- Trained on Tensorflow's Speech Command dataset

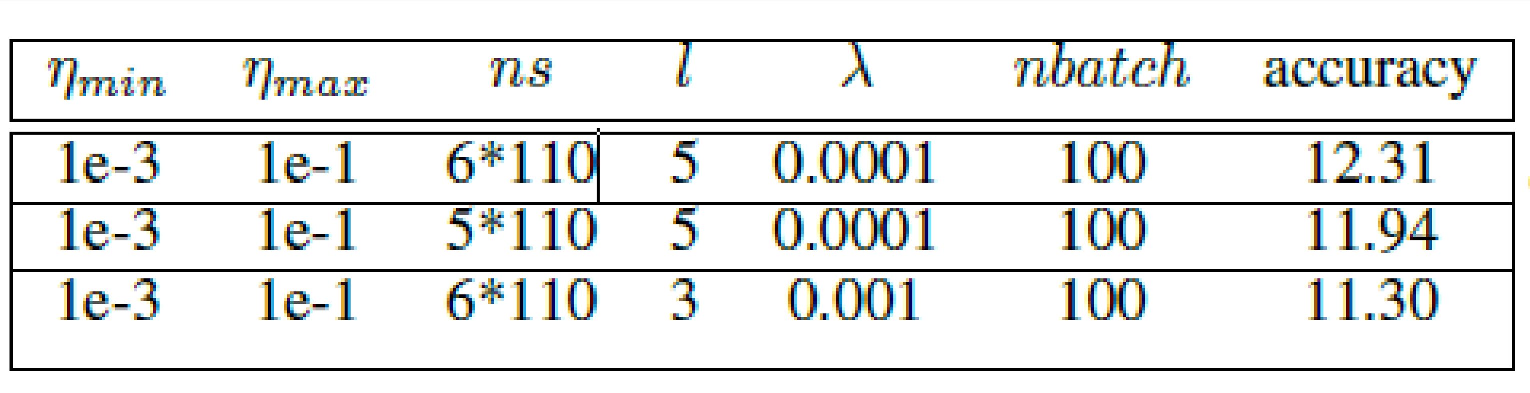




Results - Speech to text



- Accuracy on the test set was 12.31 %
- 5 Mel coefficients performed best
- Cyclic learning rate [1e-1, 1e-5]



η_{min}	η_{max}	ns	l	λ	$nbatch$	accuracy
1e-3	1e-1	6*110	5	0.0001	100	12.31
1e-3	1e-1	5*110	5	0.0001	100	11.94
1e-3	1e-1	6*110	3	0.001	100	11.30

Discussion - Speech to text

- Dataset only had 30 word classes
- Explored different number of coefficients
- Use convolutional nets instead of fully connected ones
- Explored a wide range of hyperparameters for the current implementation

Approach - Text to image

- Use pretrained word2vec model on sentence
- Modify the sentence vector to text embedding
- Train a Conditional Generative Adversarial Network on the flickr30k dataset

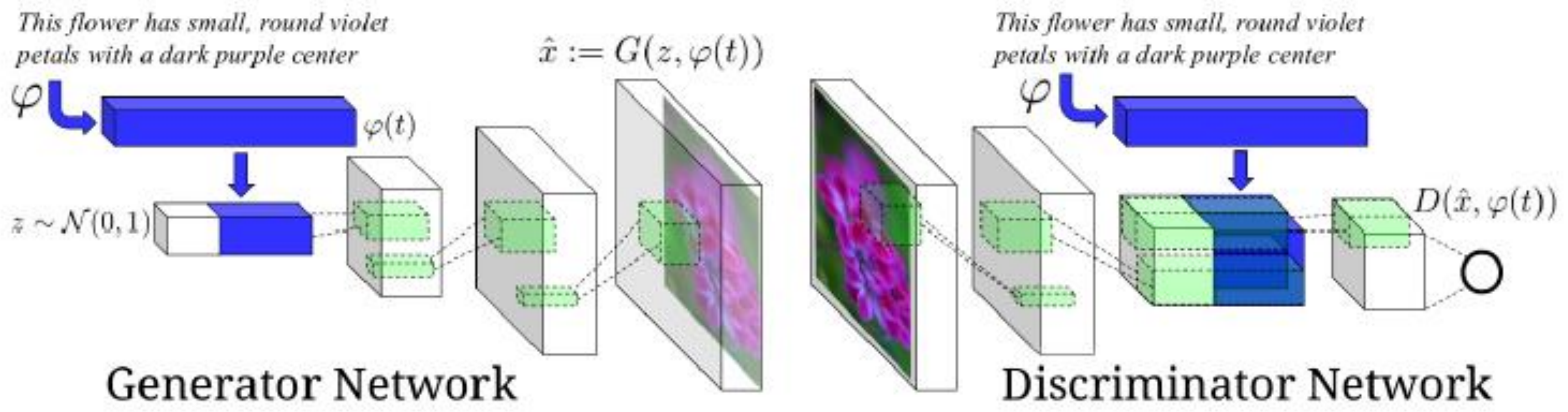
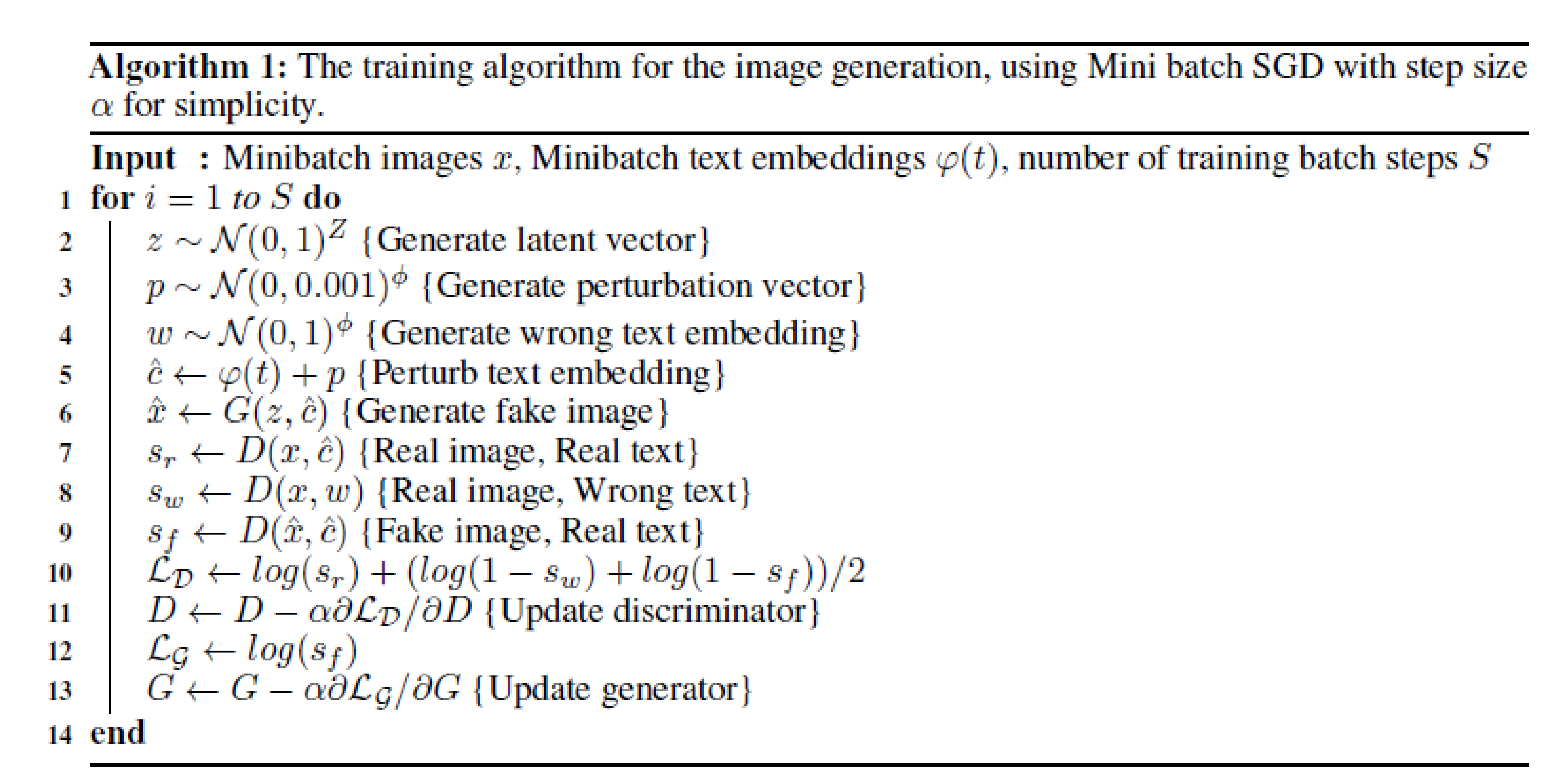

This flower has small, round violet petals with a dark purple center
φ
$\varphi(t)$
$z \sim \mathcal{N}(0,1)$
$\hat{x} := G(z, \varphi(t))$
This flower has small, round violet petals with a dark purple center
φ
$D(\hat{x}, \varphi(t))$
Generator Network
Discriminator Network

Figure 1: Overall network structure of the image generation. Image from [2]

Approach - Text to image



Algorithm 1: The training algorithm for the image generation, using Mini batch SGD with step size α for simplicity.

```
Input  : Minibatch images $x$, Minibatch text embeddings $\varphi(t)$, number of training batch steps $S$
for $i = 1$ to $S$ do
    $z \sim \mathcal{N}(0,1)^Z$ {Generate latent vector}
    $p \sim \mathcal{N}(0,0.001)^\phi$ {Generate perturbation vector}
    $w \sim \mathcal{N}(0,1)^\phi$ {Generate wrong text embedding}
    $\hat{c} \leftarrow \varphi(t) + p$ {Perturb text embedding}
    $\hat{x} \leftarrow G(z, \hat{c})$ {Generate fake image}
    $s_r \leftarrow D(x, \hat{c})$ {Real image, Real text}
    $s_w \leftarrow D(x, w)$ {Real image, Wrong text}
    $s_f \leftarrow D(\hat{x}, \hat{c})$ {Fake image, Real text}
    $\mathcal{L}_D \leftarrow log(s_r) + (log(1 - s_w) + log(1 - s_f))/2$
    $D \leftarrow D - \alpha\partial\mathcal{L}_D/\partial D$ {Update discriminator}
    $\mathcal{L}_G \leftarrow log(s_f)$
    $G \leftarrow G - \alpha\partial\mathcal{L}_G/\partial G$ {Update generator}
end
```

Results - Text to image

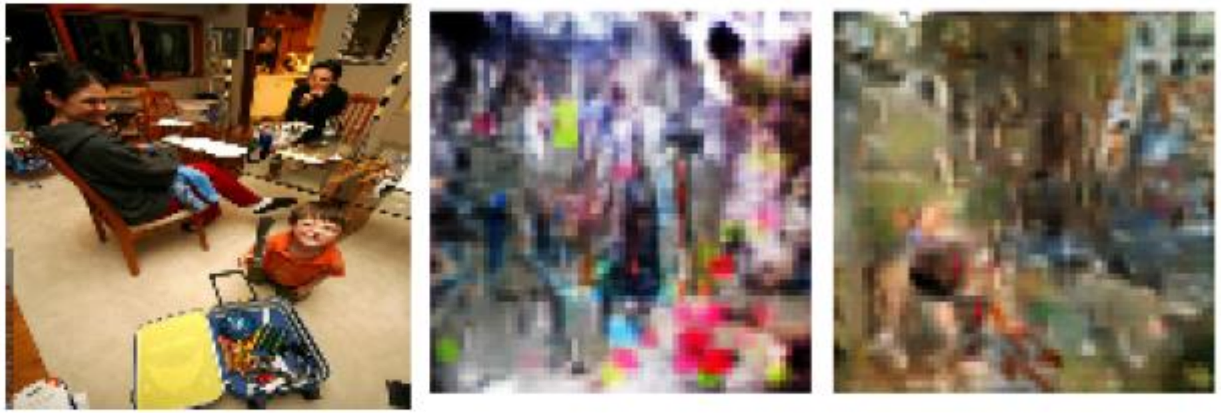


Figure 4: Caption: *A little boy shows off his suitcase full of toys.*
From left to right; Captioned Image, Output after 50 epocs, Output after 100 epochs

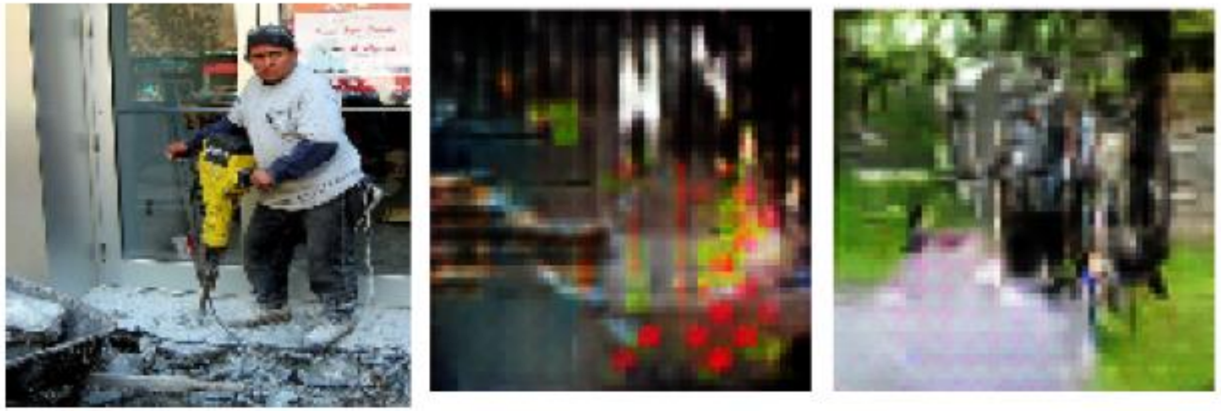
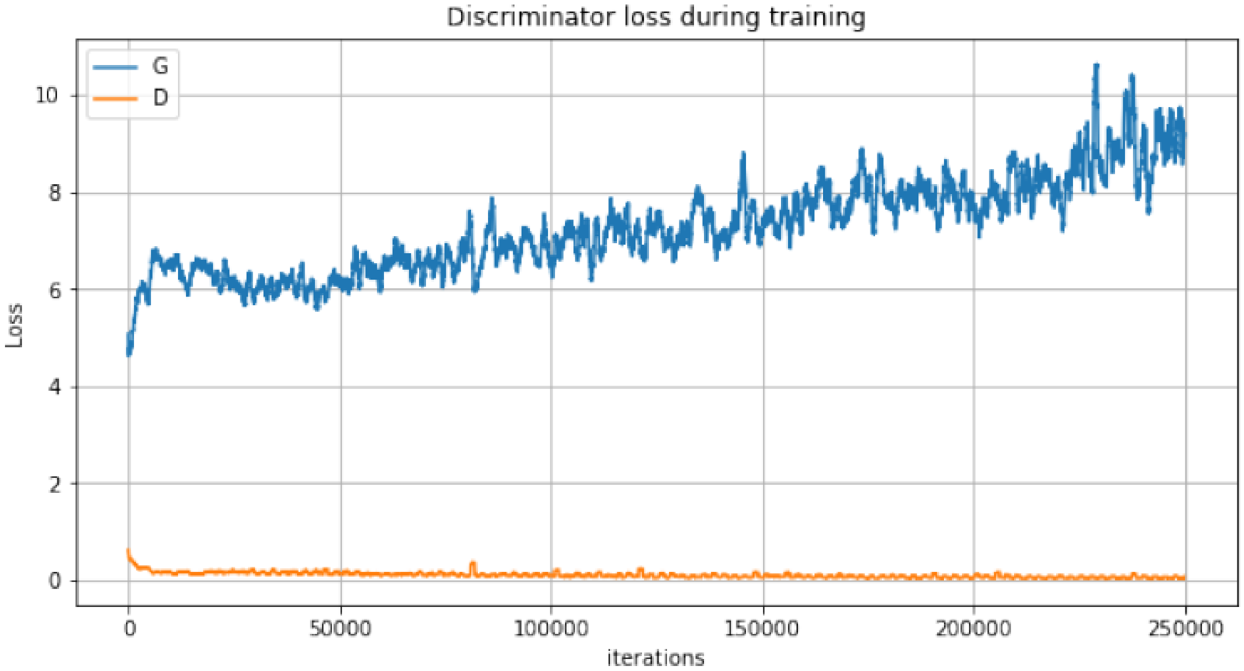


Figure 5: Caption: *A man with a jackhammer demolishing cement.*
From left to right; Captioned Image, Output after 50 epochs, Output after 100 epochs



Discussion - Text to image

- Complex scene information in the dataset
- Context aware text embedding, RNN?
- Training GANs takes a long time
- Could have extended the dataset with further condition augmentation
- Could have increased the quality with an approach like StackGAN

Conclusion - Project

- The parts performed bad independently so we didn’t focus on merging them
- Wrote all implementations from scratch in Matlab/PyTorch
- Should have focused on one particular part

Learning outcome - David

- Learned some aspects of the PyTorch framework, a lot of time was spent on just figuring out how to implement a particular thing
- Learned a lot about GANs, had never heard of them before this and read about a variety of different implementation
- Learned approaches to create feature vectors of words.

Learning outcome - Sabeen

- Learned about how to process audio files and extract them to feature vectors
- Learned basics about GAN
- Learned a lot about LSTMs despite having failed with the implementation.
- Learned to train k-layer network with speech data

Learning outcome - Qingyan

- Enjoying the process of team work and brain storming.
- Learned some text to image methods
- Learned how to complete a research, from idea finding, paper searching, coding to report writing
- Learned k-layer neural network

Thank you!