Image Deblurring - Checkpoint 1

Group 8

R. Fernández Mir C. Hawkes R. Holland
R. Hu R. Lee Mekhtieva A. Papadopoulos

1 Original Planning

The following plan is the release plan for the first sprint as we presented it in our initial *Methods and Plans* document:

Sprint	**Objectives**
09/10 - 20/10	1. Meet with the Huawei competition organisers and obtain the data set. 2. Research existing technology for classifying image blur types. 3. Research existing technology for image deblurring. 4. Ascertain which resources are required. 5. Experiment with some basic image (de)blurring techniques. 6. Set up our continuous integration tools and servers. 7. Build a simple website to output our results.

2 Organisation

As we stated in our Methods and Plans we started a GitHub repository. Not very active at the moment since the project is at a research state. Here is what the network looks like:

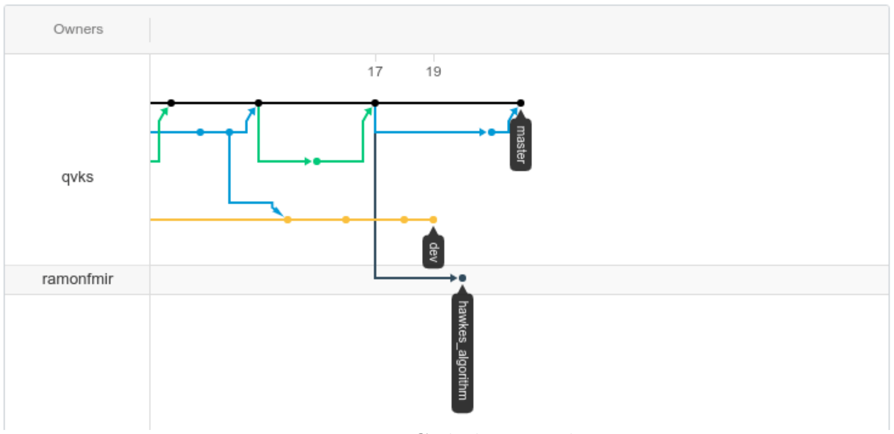
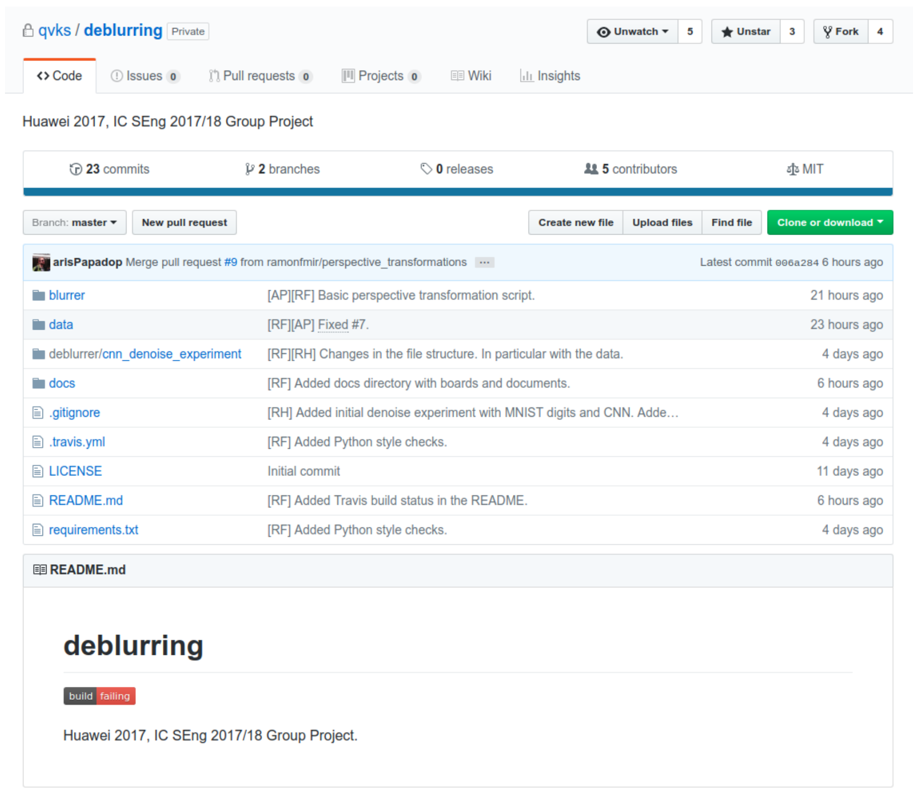


Figure 1: Github network.

And this is the current state of the repository:



Huawei 2017, IC SEng 2017/18 Group Project

arisPapadop Merge pull request #9 from ramonfmir/perspective_transformations … Latest commit 086a284 6 hours ago

blurrer	[AP][RF] Basic perspective transformation script.	21 hours ago
data	[RF][AP] Fixed #7.	23 hours ago
deblurrer/cnn_denoise_experiment	[RF][RH] Changes in the file structure. In particular with the data.	4 days ago
docs	[RF] Added docs directory with boards and documents.	6 hours ago
.gitignore	[RH] Added initial denoise experiment with MNIST digits and CNN. Adde…	4 days ago
.travis.yml	[RF] Added Python style checks.	4 days ago
LICENSE	Initial commit	11 days ago
README.md	[RF] Added Travis build status in the README.	6 hours ago
requirements.txt	[RF] Added Python style checks.	4 days ago

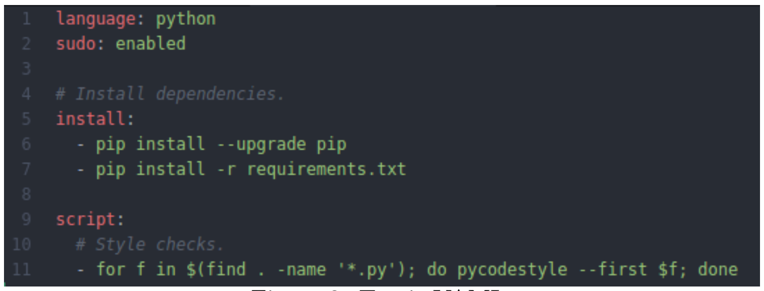
README.md

deblurring

Huawei 2017, IC SEng 2017/18 Group Project.

Figure 2: Github repositroy.

Every time we push to our git repository, the Travis build starts. We don't have any pipeline at the moment but that is our priority number one for the next sprint. For the moment, we only run Python style checks. We are not forcing them to pass, since the code we are writing is just experimental. Here are screenshots of our .travis.yml file:



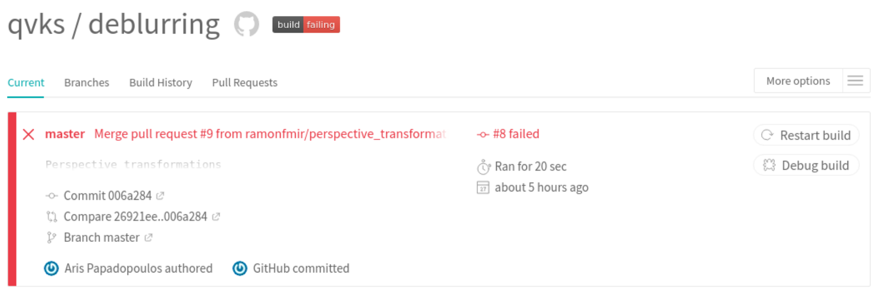
```
language: python
sudo: enabled

# Install dependencies.
install:
  - pip install --upgrade pip
  - pip install -r requirements.txt

script:
  # Style checks.
  - for f in $(find . -name '*.py'); do pycodestyle --first $f; done
```

Figure 3: Travis YAML.

Here is an example of one of our builds:



master Merge pull request #9 from ramonfmir/perspective_transformat

Perspective transformations

Commit 006a284

Compare 26921ee..006a284

Branch master

Aris Papadopoulos authored

GitHub committed

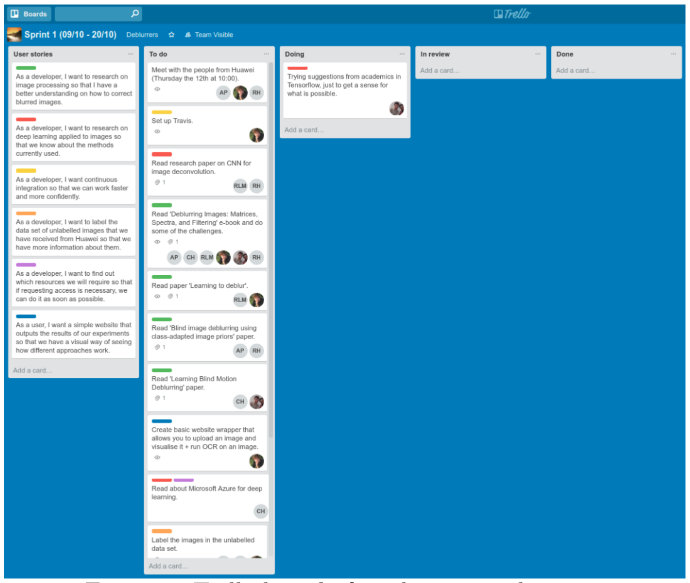
#8 failed

Ran for 20 sec

about 5 hours ago

Figure 4: Travis build.

We used Trello to organise our task board. Here are three screenshots of the state of the board at different points in time in the sprint:



Sprint 1 (09/10 - 20/10)

User stories

- As a developer, I want to research on image processing so that I have a better understanding on how to correct blurred images.
- As a developer, I want to research on deep learning applied to images so that we know about the methods currently used.
- As a developer, I want continuous integration so that we can work faster and more confidently.
- As a developer, I want to label the data set of unlabelled images that we have received from Huawei so that we have more information about them.
- As a developer, I want to find out which resources we will require so that if requesting access is necessary, we can do it as soon as possible.
- As a user, I want a simple website that outputs the results of our experiments so that we have a visual way of seeing how different approaches work.

To do

- Meet with the people from Huawei (Thursday the 12th at 10:00).
- Set up Travis.
- Read research paper on CNN for image deconvolution.
- Read 'Deblurring Images: Matrices, Spectra, and Filtering' e-book and do some of the challenges.
- Read paper 'Learning to deblur'.
- Read 'Blind image deblurring using class-adapted image priors' paper.
- Read 'Learning Blind Motion Deblurring' paper.
- Create basic website wrapper that allows you to upload an image and visualise it + run OCR on an image.
- Read about Microsoft Azure for deep learning.
- Label the images in the unlabelled data set.

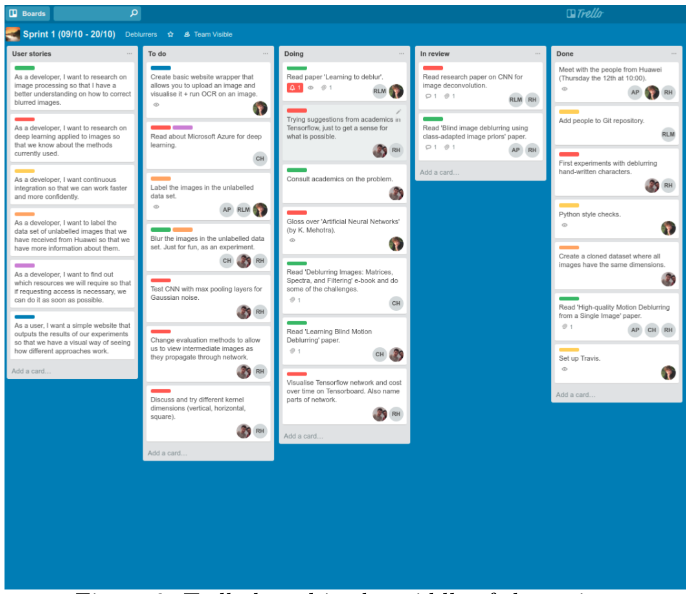
Doing

- Trying suggestions from academics in Tensorflow, just to get a sense for what is possible.

In review

Done

Figure 5: Trello board after the sprint planning.



Sprint 1 (09/10 - 20/10)

User stories

- As a developer, I want to research on image processing so that I have a better understanding on how to correct blurred images.
- As a developer, I want to research on deep learning applied to images so that we know about the methods currently used.
- As a developer, I want continuous integration so that we can work faster and more confidently.
- As a developer, I want to label the data set of unlabelled images that we have received from Huawei so that we have more information about them.
- As a developer, I want to find out which resources we will require so that if requesting access is necessary, we can do it as soon as possible.
- As a user, I want a simple website that outputs the results of our experiments so that we have a visual way of seeing how different approaches work.

To do

- Create basic website wrapper that allows you to upload an image and visualise it + run OCR on an image.
- Read about Microsoft Azure for deep learning. CH
- Label the images in the unlabelled data set. AP RLM
- Blur the images in the unlabelled data set. Just for fun, as an experiment. CH RH
- Test CNN with max pooling layers for Gaussian noise. RH
- Change evaluation methods to allow us to view intermediate images as they propagate through network. RH
- Discuss and try different kernel dimensions (vertical, horizontal, square). RH

Doing

- Read paper 'Learning to deblur'. 1 1 RLM
- Trying suggestions from academics in Tensorflow, just to get a sense for what is possible. RH
- Consult academics on the problem.
- Gloss over 'Artificial Neural Networks' (by K. Mehotra).
- Read 'Deblurring Images: Matrices, Spectra, and Filtering' e-book and do some of the challenges. 1 CH
- Read 'Learning Blind Motion Deblurring' paper. 1 CH
- Visualise Tensorflow network and cost over time on Tensorboard. Also name parts of network. RH

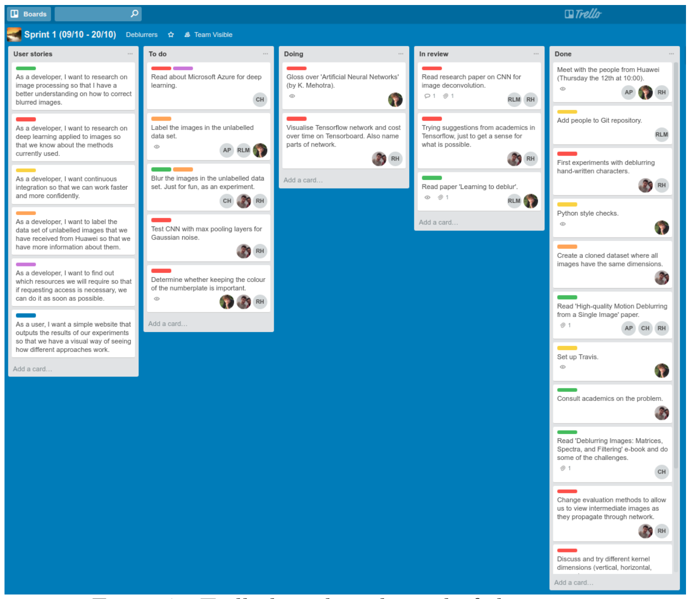
In review

- Read research paper on CNN for image deconvolution. 1 1 RLM RH
- Read 'Blind image deblurring using class-adapted image priors' paper. 1 1 AP RH

Done

- Meet with the people from Huawei (Thursday the 12th at 10:00). AP RH
- Add people to Git repository. RLM
- First experiments with deblurring hand-written characters. RH
- Python style checks.
- Create a cloned dataset where all images have the same dimensions.
- Read 'High-quality Motion Deblurring from a Single Image' paper. 1 AP CH RH
- Set up Travis.

Figure 6: Trello board in the middle of the sprint.



Sprint 1 (09/10 - 20/10)

User stories

- As a developer, I want to research on image processing so that I have a better understanding on how to correct blurred images.
- As a developer, I want to research on deep learning applied to images so that we know about the methods currently used.
- As a developer, I want continuous integration so that we can work faster and more confidently.
- As a developer, I want to label the data set of unlabelled images that we have received from Huawei so that we have more information about them.
- As a developer, I want to find out which resources we will require so that if requesting access is necessary, we can do it as soon as possible.
- As a user, I want a simple website that outputs the results of our experiments so that we have a visual way of seeing how different approaches work.

To do

- Read about Microsoft Azure for deep learning. CH
- Label the images in the unlabelled data set. AP RLM
- Blur the images in the unlabelled data set. Just for fun, as an experiment. CH RH
- Test CNN with max pooling layers for Gaussian noise. RH
- Determine whether keeping the colour of the numberplate is important. RH

Doing

- Gloss over 'Artificial Neural Networks' (by K. Mehotra).
- Visualise Tensorflow network and cost over time on Tensorboard. Also name parts of network. RH

In review

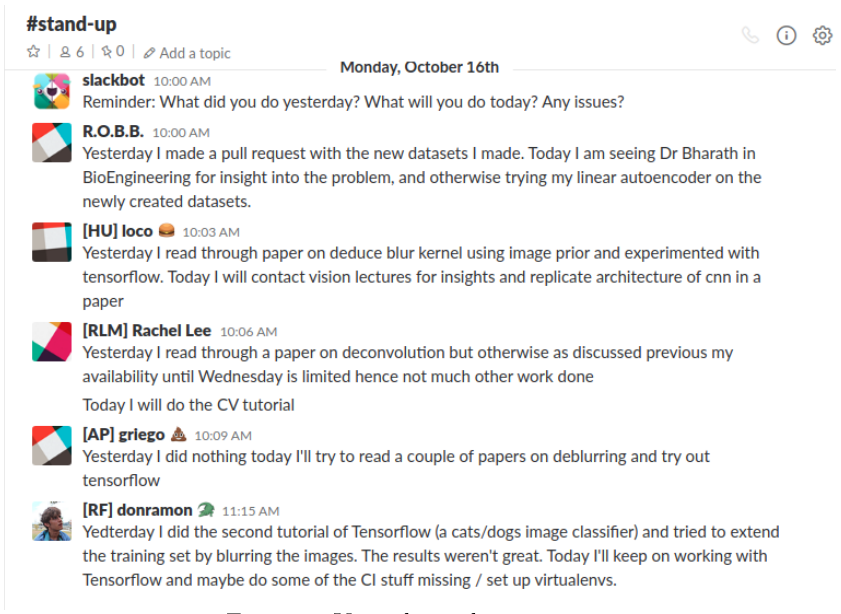
- Read research paper on CNN for image deconvolution. 1 1 RLM RH
- Trying suggestions from academics in Tensorflow, just to get a sense for what is possible. RH
- Read paper 'Learning to deblur'. 1 RLM

Done

- Meet with the people from Huawei (Thursday the 12th at 10:00). AP RH
- Add people to Git repository. RLM
- First experiments with deblurring hand-written characters. RH
- Python style checks.
- Create a cloned dataset where all images have the same dimensions.
- Read 'High-quality Motion Deblurring from a Single Image' paper. 1 AP CH RH
- Set up Travis.
- Consult academics on the problem.
- Read 'Deblurring Images: Matrices, Spectra, and Filtering' e-book and do some of the challenges. 1 CH
- Change evaluation methods to allow us to view intermediate images as they propagate through network. RH
- Discuss and try different kernel dimensions (vertical, horizontal,

Figure 7: Trello board at the end of the sprint.

Finally, we used Slack to organise our daily stand-ups in the days that not every member of the group could be present in person for a meeting at the allocated time. We present here an example of how these daily stand-ups worked:



Monday, October 16th

slackbot 10:00 AM
Reminder: What did you do yesterday? What will you do today? Any issues?

R.O.B.B. 10:00 AM
Yesterday I made a pull request with the new datasets I made. Today I am seeing Dr Bharath in BioEngineering for insight into the problem, and otherwise trying my linear autoencoder on the newly created datasets.

[HU] loco 🍔 10:03 AM
Yesterday I read through paper on deduce blur kernel using image prior and experimented with tensorflow. Today I will contact vision lectures for insights and replicate architecture of cnn in a paper

[RLM] Rachel Lee 10:06 AM
Yesterday I read through a paper on deconvolution but otherwise as discussed previous my availability until Wednesday is limited hence not much other work done

Today I will do the CV tutorial

[AP] griego 💩 10:09 AM
Yesterday I did nothing today I'll try to read a couple of papers on deblurring and try out tensorflow

[RF] donramon 🐊 11:15 AM
Yedterday I did the second tutorial of Tensorflow (a cats/dogs image classifier) and tried to extend the training set by blurring the images. The results weren't great. Today I'll keep on working with Tensorflow and maybe do some of the CI stuff missing / set up virtualenvs.

Figure 8: Virtual stand up meeting.

3 Tasks Accomplished

3.1 Sprint Overview

Keeping the numbering of the tasks planned for the first sprint from the table in page 1, in this sprint we did the following:

For 1, a meeting between the head of the Machine Learning Department of Huawei and the two teams which are implementing the same project, namely group 8 and group 16 , was arranged on the 12th of October. In the meeting various aspects of the competition were discussed and we have kept in touch with Huawei for additional information on the project.

For 2 we read a list of papers and books on the subject matter, these include the following:

- Deblurring Images: Matrices, Spectra, and Filtering
- High-quality Motion Deblurring from a Single Image
- Learning to Deblur

This list is incomplete, but these are the main papers on Image Processing that have influenced the approach we will take in the next sprint. We have also read some papers about neural networks for deblurring, from which we would highlight Deep Convolutional Neural Network for Image Deconvolution.

For 3 we tried out various existing tools to get a sense of the overall quality of existing implementations and a better understanding of what our results should be like. These tools include:

- MATLAB. It was useful for the initial experiments with image processing.
- OpenCV. We learned about this very powerful Computer Vision library, in particular about the image processing part of it.
- TensorFlow. We did some of the tutorials and played with it by building our own naive CNN to deblur images.

The research described above lead us to the following conclusions, regarding the tools we would require to use for the project, as stated in 4:

- Image processing is not enough to tackle this problem. The images are very small and noise plays an important role so we cannot use techniques based on a single image.
- We will use image processing techniques to augment the data set and also as an aid in the training stage. All of these will be done using OpenCV.
- A convolutional neural network will be the core of our algorithm for deblurring. And we will use TensorFlow to build it. It is essential to train the network using different kinds of blur, detailed in 5, due to the differences in noise distribution in varying types of blur. In section 3.2 we explain the current state of our neural network.
- Even though this is our initial approach we want to try out other more adventurous ideas and see if they lead to better results. Some of these are splitting the alphanumerical sequences and training with characters or auto-generating high quality license plates as training data.

For 5 we experimented with various image blurring techniques.

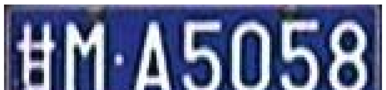
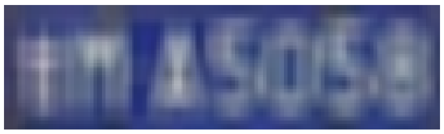
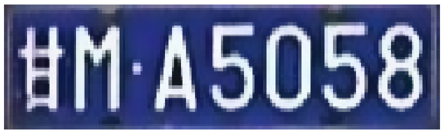


Figure 9: Original Image.

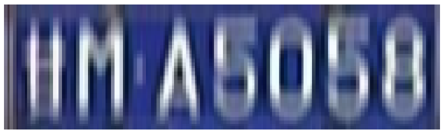


(a) Averaging blur.

(b) Median blur.



(c) Bilateral blur.



(d) Motion blur.

Figure 10: Different Blurs

For 6 we set up the TravisCI continuous integration server which for nows only runs Python style checks every time someone pushes to master.

As far as 7 is concerned we decided to shift our approach from a website to an Android application. We present here some of the mock-ups:

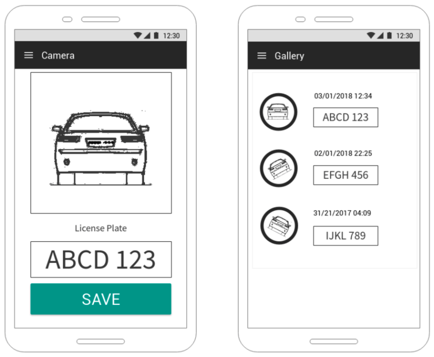
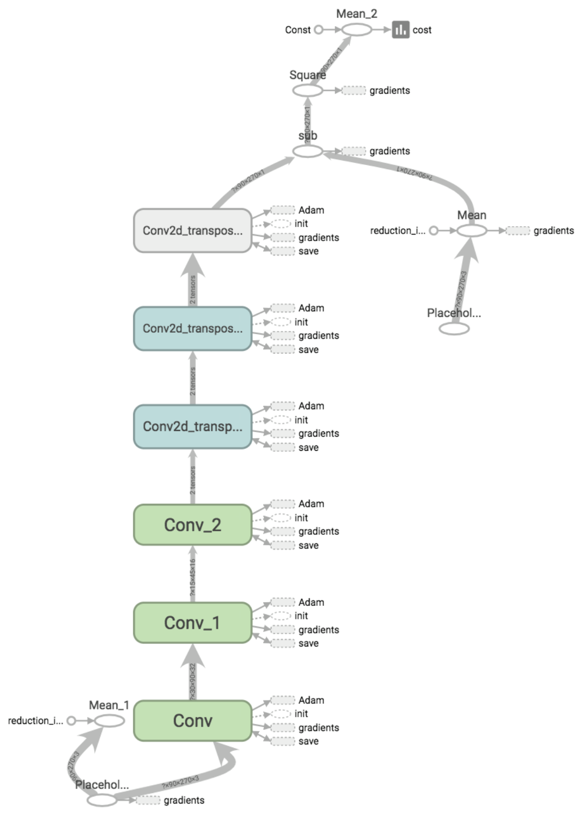


Figure 11: Android app mockups.

3.2 Current CNN Iteration

Below is our first attempt to build a CNN for image deblurring in Tensorflow. This graph was generated by Tensorboard; a tool we found during our research. It was extremely useful in debugging out implementation of the network.



This defines our skeleton for the state-of-the-art autoencoding CNN. We start by applying a Guassian blur to our original image and then converting it to grey scale (individual colour channels contain little additional information, and any reduction in dimensions reduces training time). Then the image is propagated through three sequential convolution layers, each with a set number of kernels with varying strides and dimensions. Similarly, it is then passed through three transpose convolutional, or *deconvolutional*, layers. The resulting image can then be compared with the original image, a cost (MSE) calculated and the kernel weights can be adjusted appropriately with Tensorflow's Adam optimizer.

3.2.1 Data preparation

To train our CNN a stream of data is required. Our current data preparation system loads the entire unblurred image data into an array and blurs them using our blur model. This is fed into the CNN in batches. This current system does not use modern techniques such as random shuffling the array which will benefit the learning rate. As our sample data sets are images of different size, we standardise the size of image to 270*90 pixels by padding the expanded space with black. This will keep the aspect ratio of the image and prevent blurring from preprocessing.

3.2.2 Results

This CNN is a skeleton trained on CPU, and does not take advantage of max-pooling and other beneficial architectures. After training on the data set for a number of iterations of the dataset we obtain the following results:

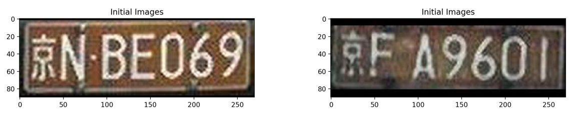


Figure 12: Original clean image from train set

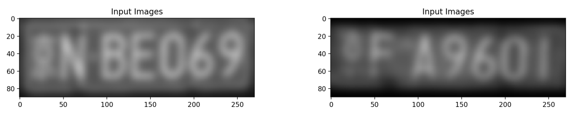


Figure 13: Blurred and grey scaled input to network

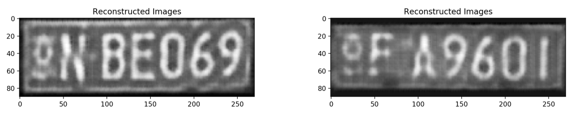
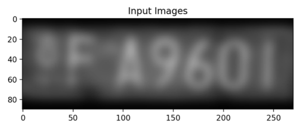
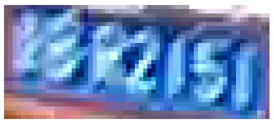


Figure 14: Network output

These are the results that we expected to reproduce in Sprint 1. The simplistic / experimental CNN has effectively learnt deblurring filters that we could have created ourselves. For sprint 2 we plan to extend this model and train it with a collection of different noising techniques. For example, it should be noted that the applied Guassian blur does not accurately mimic images blurred and noised in the real world. The comparison below illustrates this discrepancy:



(a) Simple blur method we used.



(b) Real Life example.

Figure 15: Simplistic / Unrealistic Blurs

4 Project Plan

We will have our sprint planning meeting on Monday, but these are the main objects we want to achieve:

1. Have an end-to-end pipeline that keeps track of performance metrics of every build so that we know which direction to move to.
2. Have a basic Android app to take a picture and process it.
3. Keep working on our current neural network and try out different blurring techniques on the training data to make it better.
4. As mentioned before, try riskier approaches and see if they lead to good results.