# Deep Sentiment Analysis on Tumblr

**Anthony Hu**

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Master of Science in Applied Statistics


Department of Statistics
University of Oxford
Oxford, United Kingdom

September 2017

# Declaration

The work in this thesis is based on research carried out at the Department of Statistics, University of Oxford. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

# Acknowledgements

I would like to thank my supervisor Seth Flaxman who always had great insights and ideas. I would also like to thank my parents for giving me the opportunity to study in Oxford, and for their unfaltering support.

# Deep Sentiment Analysis on Tumblr

**Anthony Hu**

Submitted for the degree of Master of Science in Applied Statistics
September 2017

## Abstract

This thesis proposes a novel approach to sentiment analysis using deep neural networks on both image and text. Deep convolutional layers extract relevant features on Tumblr photos and high-dimensional word embedding followed by a recurrent layer process the textual information to accurately infer the emotion of the post. The network architecture, named Deep Sentiment, can also be adapted to generate images and text given an emotion.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Sentiment analysis has been an active area of research in the past few years, especially on the readily available Twitter data, e.g. Bollen et al. [2] who investigated the impact of collective mood states on stock market or Flaxman et al. [1] who analysed day-of-week population well-being.

Contrary to Twitter, Tumblr's posts are not limited to 140 characters, allowing more expressiveness, and are not centered on the textual content but on the image content instead. A Tumblr post will almost always be an image with some text accompanying the latter. Pictures have become prevalent on social media and characterising them could enable the understanding of billions of users.

http://www.ifp.illinois.edu/ jyang29/papers/AAAI15-sentiment.pdf

We propose a novel method to uncover the emotional of an individual posting on social media. The ground truth emotion will be extracted from the tags, considered as the 'self-reported' emotion of the user. Our model incorporates both text and image and we aim to 'read' them to be able to understand the emotional content they imply about the user.

# Chapter 2

# Tumblr Data

## 2.1 Overview of the Data

Tumblr posts were retrieved using the Python API, here is an example of a post:


Figure 2.1: An example of a Tumblr post

The tags '#chowchow # home #happy #bluetongue' are really valuable as they indicate the user's state of mind when writing that post. Ekman popularised the idea that there are six basic emotions [3]: hapiness, sadness, anger, surprise, fear, disgust. These emotions are said to be *basic* as they are hardwired regardless of the species. Basic emotions are innate, universal and automatic and induce fast reactions that are linked with a high survival rate.

To build our dataset, queries were made searching for each of the six emotions appearing in the tags. Adjectives were used as they were more commonly used by users: #happy, #sad, #angry, #surprised, #scared and #disgusted. Each post would then contain the following information:

1. The text, in the example above: *When dogs are back home!*
2. The picture.
3. The associated emotion: one among the six classes.

Note that sometimes, a post would contain several basic emotions such as '#sad #angry'. We simply selected the first hashtag written by the user as it can be deemed as the main emotion that the user first thought of.

The data extraction took several weeks due to the API's limitations: 1,000 requests per hour and 5,000 requests per day, with each request containing 20 posts. The final dataset has about 1 million posts.

## 2.2 Data Preprocessing

In some posts, the tag also appeared in the text itself, for instance:

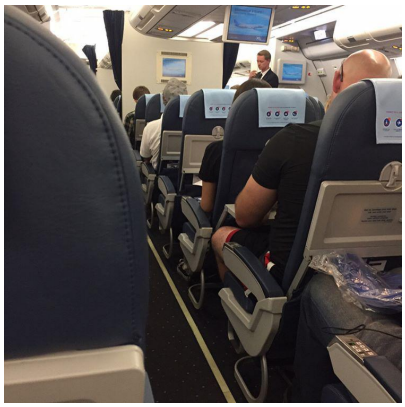
> *"When you're on vacations and there is a rainstorm. #fail #sad".*

Keeping the *#sad* would bias the learning process and the neural network would simply learn to detect the presence or absence of that tag. To ensure that the network is actually learning something, we removed the hashtags containing the emotion to be predicted.

Also, Tumblr is used worldwide, therefore posts not written in English had to be removed from the training data. Basically, if a post contains less than $t$, a threshold, English word, it is deemed as non-English and removed from the dataset. The threshold was set to 5 English words as it appears to filter out reasonably well the dataset. The vocabulary of english words was obtained from Word2Vec and will be detailed further in Section 4.

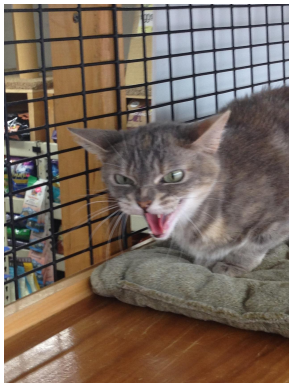
Here are examples of posts with their associated emotions:

(a) **Happy**: "Just relax with this amazing view #bigsur #california #roadtrip #usa #life #fitness (at McWay Falls)"

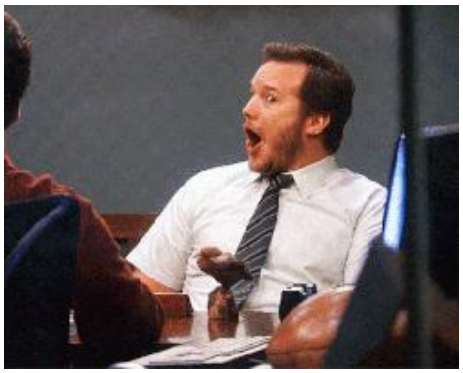


(b) **Scared**: "On a plane guys! We're about to head out into the sky to Paris, France #Paris #trip #kinda #nervous #fun #vacations"


(c) **Sad**: "It's okay to be upset. It's okay to not always be happy. It's okay to cry. Never hide your emotions in fear of upsetting others or of being a bother If you think no one will listen. Then I will."



(d) **Angry**: "Tensions were high this Caturday..."



(e) **Surprised**: "Which Tea? Peppermint tea: What is your favorite gif right now?"

(f) **Disgusted**: "Me when I see a couple expressing their affection in physical ways in public"

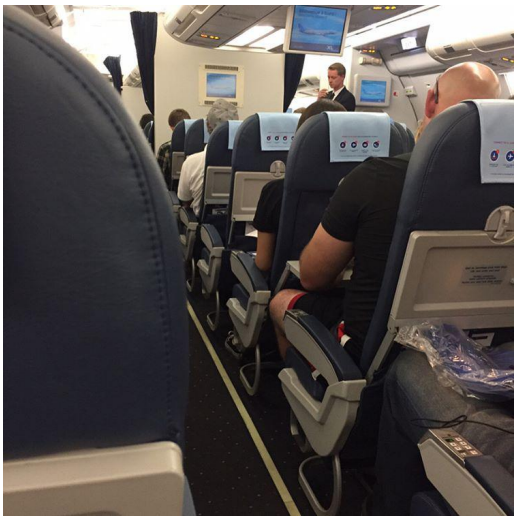
Figure 2.2: The 6 emotions illustrated by Tumblr posts [4]

# Chapter 3

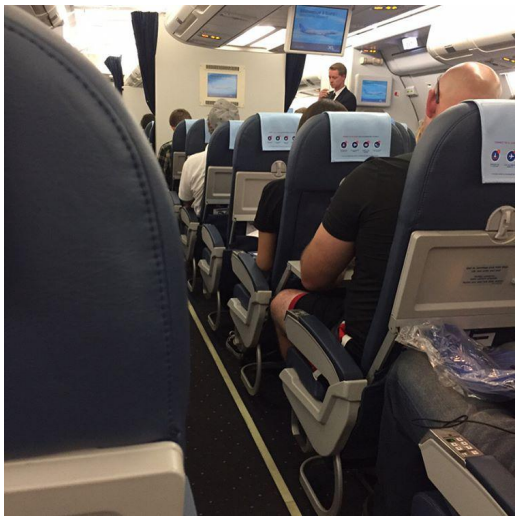
# Deep Sentiment

Real-world information oftentimes come in several modalities. For instance, in speech recognition, humans integrate audio and visual information to understand speech, as was demonstrated by the McGurk effect [26]. Separating what we see from what we hear seems like an easy task, but in the experiment conducted by McGurk, the subjects who were listening to a /*ba*/ sound with a visual /*ga*/ actually reported they were hearing a /*da*/. This is uncanny as even if you know the actual sound is a /*ba*/, you cannot stop your brain from interpreting it as a /*da*/.

Likewise, an image almost always come with a text as different interpretation can arise when a textual context is not provided, as shown in Figure 3.1:



(a) "Planes might just be the most frightening thing ever." **scared**



(b) "I hate it when people are taking too much space on planes." **angry**

Figure 3.1: Different meanings with different captions.

Exploiting both visual and textual information is therefore key to understand the user's emotional state. Deep Sentiment is the name of the deep neural network incorporating visual recognition and text analysis.

## 3.1 The Architecture

Deep Sentiment builds on the models we have seen before as shown in Figure 3.2:

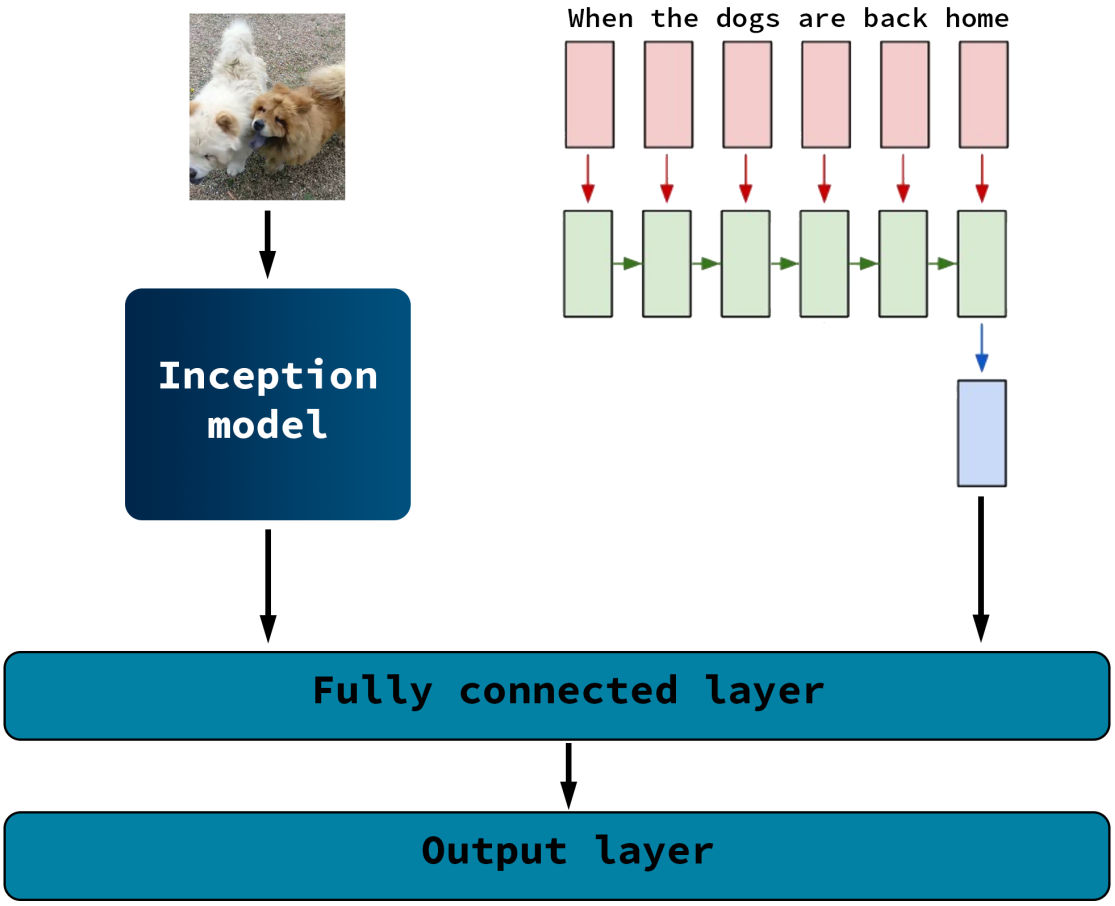



Figure 3.2: Deep Sentiment architecture

1. The image go through the pre-trained Inception model that will extract features from the images, more precisely with 128 neurons in the last Inception layer.

2. The text is embedded in a high-dimensional space with Word2Vec and will be fed to an LSTM with a 2048 neurons output layer.

3. The two outputs are concatenated to form the 'Fully connected layer' in the graph with 2176 neurons.

4. The final layer with 6 neurons one for each basic emotion.

5. A softmax layer to give the probability distribution of the emotional state of the user.

## 3.2 Results

Deep Sentiment was trained with:

- 10,000 steps
- Mini-batch size of 32
- Adam optimizer with initial learning rate of 1e−3
- Learning rate decay of $\frac{1}{2}$ every 1000 steps

The training process of the Inception fine-tuning was monitored thanks to Tensorboard:

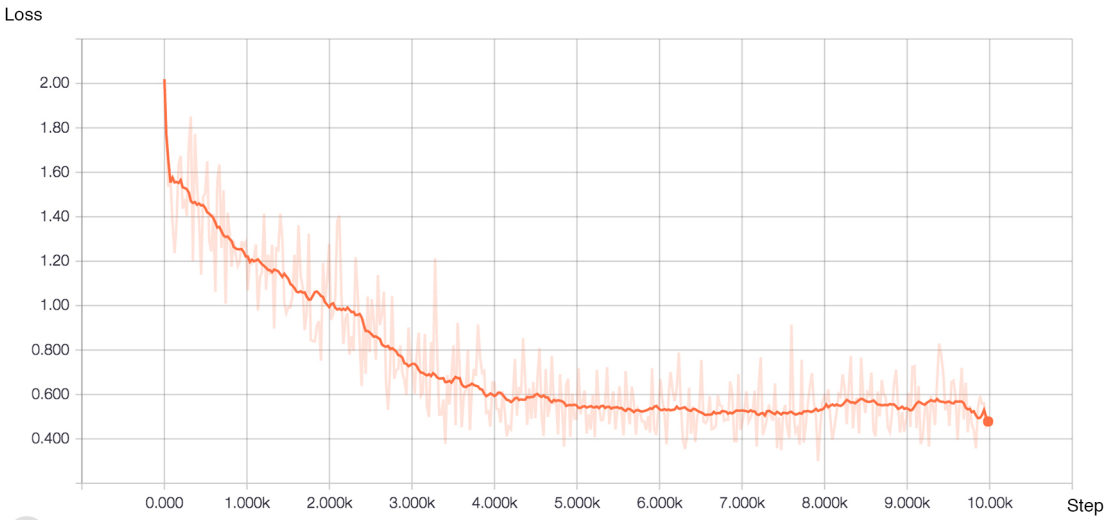



Figure 3.3: Loss function of Deep Sentiment

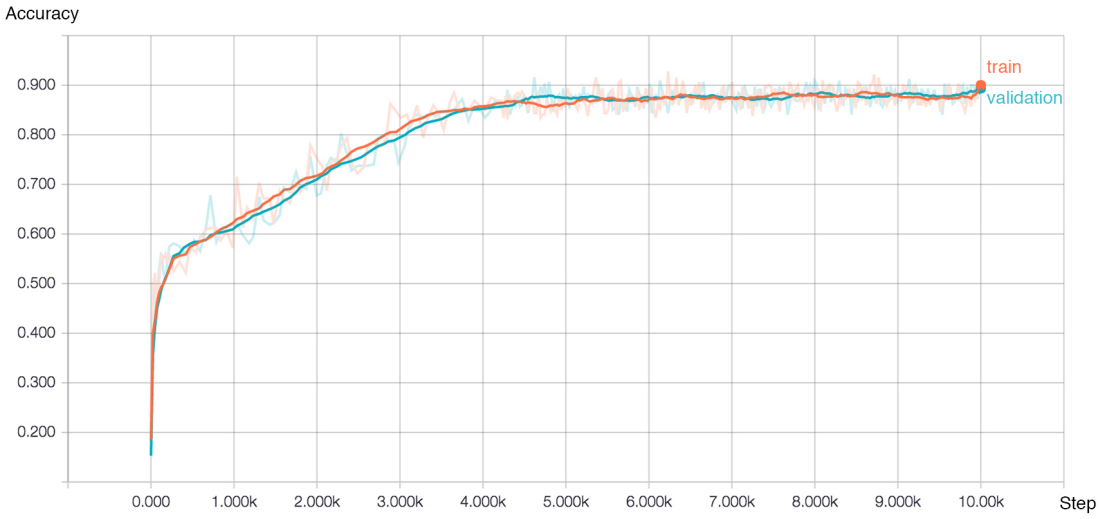



Figure 3.4: Train/validation accuracies of Deep Sentiment

graph with the other models as well.

90% train accuracy and 89% test accuracy!

## 3.3 Emotion visualisation

We can generate an image that maximises the score of a certain emotion by performing gradient ascent on a randomly initialised image [27].

More concretely, let $I$ be an image and $y$ be a target emotion. Let us denote by $s_y(I)$ the score of class $y$ for the image $I$, that is one of the six neurons right before the softmax layer. We want to generate an image with a high score for emotion $y$ by solving the problem:

$$I^* = \arg\max_I \; s_y(I) - R(I) \tag{3.1}$$

with $R(I)$ a regulariser that contains both explicit and implicit regularisation we will describe shortly.

Note that we're maximising the unnormalised class scores $s_y(I)$ and not the probabilities returned by the softmax: $\frac{s_y(I)}{\sum_c s_c(I)}$. The reason is that maximising the softmax probabilities can be achieved by minimising the scores of the other emotions. Instead, we want to make sure the optimisation concentrates on the emotion we want to visualise.

### 3.3.1 Regularisation

The explicit regulariser is the $L_2$ decay: $R(I) = \lambda \|I\|_2^2$ that prevents extreme pixel values from dominating the generated image. Those pixel values do not occur naturally in real images and are not useful for visualisation.

The implicit regularisations are [28]:

1. **Gaussian blur:** Gradient ascent tends to produce image with high frequency information. What are frequencies in images? To put it simply, each image is made of various frequencies: start with the average colour (low frequency) and slowly add higher frequencies wavelengths to build the details of the image.

   An image with high frequency information causes high activations but are not realistic nor interpretable as shown by Nguyen et al. [29]. High frequency information are penalised using a Gaussian blur step on image $I$ :

GaussianBlur$(I, \theta_{\text{blur}})$ with $\theta_{\text{blur}}$ the standard deviation of the Gaussian kernel used in the blur step. Blurring an image is computationally expensive and as such, we're only blurring every $\theta_{\text{blur-every}}$ steps.

2. **Pixel clipping:** After performing $L_2$ decay and Gaussian blur, that suppress high amplitude and high frequency information, we're left with images with pixel values that are small and smooth. However, each pixel will still be non-zero and contribute a little bit to the gradient. We want to discard the contribution of unimportant pixels and focus only on the main object. That can be done by setting pixels with small norm (over the red, green, blue channels) to zero. The threshold $\theta_{\text{small-norm}}$ for the norm is set to be a percentile of all pixel norms in the image.

### 3.3.2 Generated images

We performed gradient ascent on a randomly initialised image using the following parameters:

- $L_2$ regularisation parameter: $\lambda = 0.001$.
- In Gaussian blur, $\theta_{\text{blur}} = 0.5$ and $\theta_{\text{blur-every}} = 10$.
- In pixel clipping, $\theta_{\text{small-norm}}$ is the norm of the $10^{\text{th}}$ percentile.
- 500 gradient updates.

Maximising over the emotion 'happy' yielded:

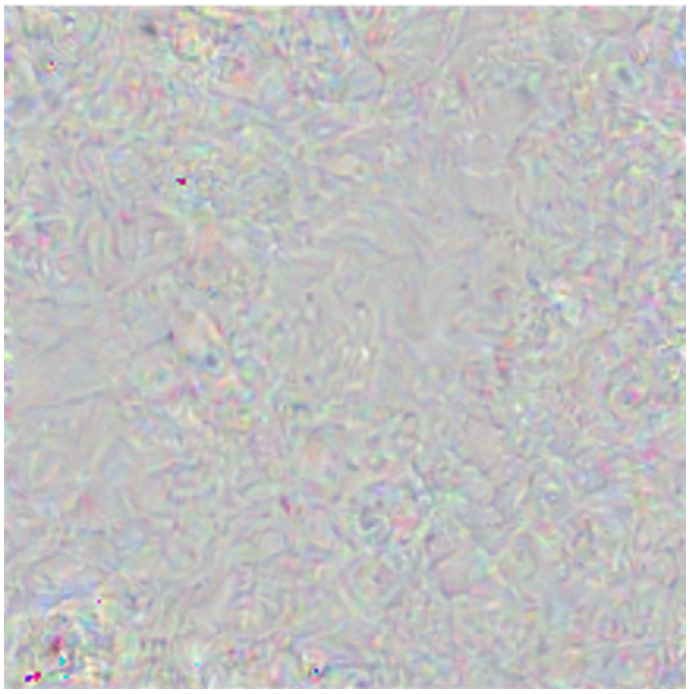


Figure 3.5: Generated image maximising happiness

## 3.4 Generate text posts

We can tweak Deep Sentiment to instead make the neural network generate text by feeding an image. The network will be trained to predict the next word of the text as shown in Figure **??**:

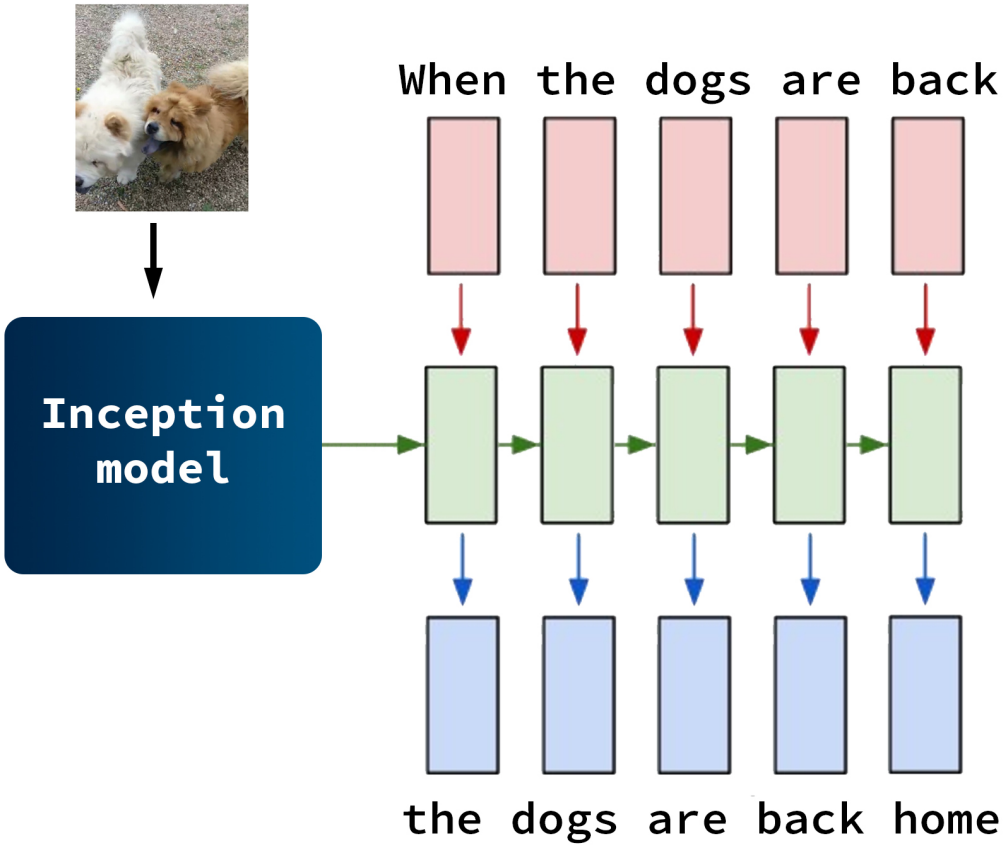



Figure 3.6: Deep Sentiment for text generation

# Bibliography


[1] S. Flaxman and K. Kassam, On #agony and #ecstasy: Potential and pitfalls of linguistic sentiment analysis. In preparation, 2016.

[2] J. Bollen, H. Mao, X.-J. Zeng, Twitter mood predicts the stock market. In *Journal of Computational Science*, 2011.

[3] P. Ekman, An Argument for Basic Emotions. In *Cognitive and Emotion*, 1992.

[4] Tumblr photos:
http://fordosjulius.tumblr.com/post/161996729297/just-relax-with-amazing-view-ocean-and
http://ybacony.tumblr.com/post/161878010606/on-a-plane-bitchessss-we-about-to-head-out
https://little-sleepingkitten.tumblr.com/post/161996340361/its-okay-to-be-upset-its-okay-to-not-always-be
http://shydragon327.tumblr.com/post/161929701863/tensions-were-high-this-caturday
https://beardytheshank.tumblr.com/post/161087141680/which-tea-peppermint-tea-what-is-your-favorite
https://idreamtofflying.tumblr.com/post/161651437343/me-when-i-see-a-couple-expressing-their-affection

[5] D. H. Huble and T. N. Wiesel, Receptive fields and functional architecture of monkey striate cortex. In *Journal of Physiology (London)*, 1968.

[6] Convolution images, M. Gorner, Tensorflow and Deep Learning without a PhD, Presentation at *Google Cloud Next '17*:

https://docs.google.com/presentation/d/1TVixw6ItiZ8igjp6U17tcgoFrLSaHWQmMOwjlgQY9co/pub?slide=id.g1245051c73_0_2184
The slide on the convolutional neural network was adapted to our architecture.

[7] V. Nair and G. E. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML*, 2010.

[8] Max pooling image, Cambridge Spark: https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html

[9] A. Krizhevsky, I. Sutskever and G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.

[10] C. Szegedy et al., Going deeper with convolutions. In *CVPR*, 2015.

[11] K. He et al., Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[12] A. Karpathy, L. Fei-Fei, J. Johnson, Transfer Learning. In *Stanford CS231n Convolutional Neural Networks for Visual Recognition*, 2016.

[13] S. Arora et al., Provable Bounds for Learning Some Deep Representations. In *ICML*, 2014.

[14] D. Hebb, in his book The Organization of Behavior, 1949.

[15] Video explaning Inception Module, https://www.youtube.com/watch?v=VxhSouuSZDY.

[16] T. Mikolov et al., Efficient Estimation of Word Representations in Vector Space. In *ICLR*, 2013.

[17] TensorFlow, Word2Vec tutorial, https://www.tensorflow.org/tutorials/word2vec.

[18] C. McCormick, Word2Vec Tutorial - The Skip-Gram Model, http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/.

[19] T. Mikolov et al., Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*, 2013.

[20] A. Mnih and Y. W. Teh, A fast and simple algorithm for training neural probabilistic language models. In *ICML*, 2012.

[21] B. Zoph et al., Simple, Fast Noise-Contrastive Estimation for Large RNN Vocabularies. In NAACL, 2016.

[22] Word2Vec pre-trained model, Google, 2013. https://code.google.com/archive/p/word2vec/

[23] S. Flaxman et al., Who Supported Obama in 2012? Ecological Inference through Distribution Regression. In *KDD*, 2015.

[24] A. Karpathy, The Unreasonable Effectiveness of Recurrent Neural Networks, 2015. http://karpathy.github.io/2015/05/21/rnn-effectiveness/

[25] C. Olah, Understanding LSTM Networks, 2015. http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[26] H. McGurk and J. MacDonald, Hearing lips and seeing voices. In *Nature*, 1976.

[27] K. Simonyan, A. Vedaldi, and A. Zisserman, Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *ICLR*, 2014.

[28] J. Yosinski et al., Understanding Neural Networks Through Deep Visualization. In *ICML* 2015.

[29] A. Nguyen et al., Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. In *CVPR*, 2015.