Author Profiling based on Text and Images
Notebook for PAN at CLEF 2018

Luka Stout, Robert Musters, and Chris Pool

Anchormen, The Netherlands
{l.stout,r.musters,c.pool}@anchormen.nl

Abstract In this paper we describe our participation in the PAN 2018 shared task of Author Profiling. In this task we identify the gender of authors based on written text and shared images. We describe our approaches to the text-based, image-based and the combined task. The presence of three different languages raises the question whether a single model architecture can be built that works well on all three languages. We also propose a way to combine multiple predictions on shared content into a single prediction on user-level. Our final system for text is an ensemble of a Naive Bayes model and a RNN with attention. The image classification is done by finding selfies and predicting the gender of the person on those images using CNNs.

1 Introduction

With the gaining influence and importance of social media it becomes more and more relevant to gain insights into the authors of content, mostly made up of images and text. Because social media networks allow people to create anonymous accounts it becomes of greater interest to the research community to get to know users on social media. Knowing specific details about a user, like gender, age, native language or emotional state is an interesting challenge for the marketing, forensic and security sectors. Author Profiling[1] is the task of determining an author's features like gender, age, language variety by understanding their online persona. In addition to the tweets the shared task of 2018[2] includes images that were shared by the authors as well. The goal is to infer the gender of an author given one-hundred of their tweets and ten images, in three different languages, English, Spanish and Arabic. The presence of three different languages raises the question whether a single model architecture can be built that works well on all three languages. The shared task is divided into three subtasks: Infer the gender based on tweets, based on their shared images and a combination of the two. We have focused on the text-based task, however we have also developed an image-based approach to also participate in the combined task. For this we experimented with traditional techniques, such as tf-idf and Naive Bayes[3], as well as deep learning techniques, such as Recurrent Neural Networks (RNNs)[4] and Convolutional Neural networks (CNNs)[5]. In this paper we describe our final systems and results.

2 Dataset Description and Preprocessing

The PAN 2018 Author Profiling[6] training set consists of text in three different languages and images grouped by authors, who are labeled by gender and language. The number of authors per gender is balanced in every language. This training set was used for feature engineering, parameter tuning and training of the classification model. For the languages English, Spanish and Arabic we received a dataset containing 100 tweets and 10 images per author. For English and Spanish there are 3,000 authors and for Arabic 1,500 authors. This gives a total of 750,000 tweets and 75,000 images. The goal of the task is to predict the gender of a user given these 100 tweets and 10 images. We have chosen to create models on tweet and image level and combine the predictions to create a single prediction for every author.

The following preprocessing steps were performed, the two additional preprocessing steps for Arabic can be found online[*]:

- Replaced numbers, URLs, hashtags, mentions, emojis and smileys with their own unique tokens.
- Used a tokenizer to filter out punctuation and tokenize sentences into a list of lowercase words.
- Expanded contractions. (For English)
- Normalization of tokens, namely unifying the orthography of *alifs*, *hamzas*, and *yas*/*alif maqsuras*. (For Arabic)
- Noise removal, i.e. removing short vowels and other symbols (*harakat*). (For Arabic)

After preprocessing and tokenization, the maximum number of words in a tweet is 39 for English. For the other languages there are fewer than 200 tweets longer than 39 words. As this only accounts for 0.02% of all the tweets in the data set and to keep the models consistent across languages, we have decided to cap the number of words in a sentence to 39.

Basile et al.[7] note that augmenting the tweet dataset with the data of previous Author Profiling tasks[8, 9] does not improve the performance of the resulting classifiers. They emphasize that this is due to temporal differences in the data. We have seen that topics reflect events from 2017 are definitely present in the data. While the data from previous years contains data with events from 2016 and before. As such we have decided not to include additional datasets to limit the effects of these differences.

For the image classification task we have used additional data to create our classifier: a selfie dataset[10] and the MIRFLICKR dataset[11]. Their use is explained in Section 5.

3 Prediction Strategies

There are two ways to predict gender based on an author's social media content. The first is to treat all of the content as a single item and create a single prediction based on the entirety of the data. This is analogous to a bag of words approach in the case of text. However, concatenating or summing up the images at pixel level is not straightforward and does not make intuitive sense. As such we have chosen a different approach. The approach is to make predictions on the item-level and combine these predictions somehow.

[*] https://maximromanov.github.io/2013/01-02.html

There are multiple ways of constructing an author-level prediction based on tweet-level predictions, whether it is text, images or a combination of both. We have used three different strategies. (1) The first strategy is using the majority class of all predictions. (2) The second strategy is to use the mean probability of all predictions. (3) The last strategy is to only use predictions where the model is very sure that an input indicates a certain gender. With the latter strategy a weighted average of the predictions where the weights are zero for predictions that are within a certain range is used:

$$w_i = \begin{cases} 0 & \text{if } \alpha < P_i(female) < \beta, \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

$$P(female) = \frac{1}{N} \sum_{i=1}^{N} w_i P_i(female) \tag{2}$$

where P is the prediction by a single model for a single author, P_i is the prediction for the i-th tweet or image of the author and N the number of tweets or images for the author. If no such prediction exists we fall back to the second strategy, the mean strategy. We have found $\alpha = 0.25$ and $\beta = 0.75$ to be good default values. Our usage of the third prediction strategy improved our accuracy on a validation set, as illustrated in Table 1. The rationale is that the predictions where the model is sure that a certain input points towards a specific gender are the only ones that have any influence on the author-level prediction.

	acc		σ
(1) Majority	0.767	±	0.003
(2) Mean	0.780	±	0.011
(3) Sure	0.790	±	0.008

Table 1. 3-fold validation accuracy of using the Recurrent Neural Network on the English text. The model was trained on tweet level and then the strategies were applied to a tweet set of unseen authors. This gives an example of the performance increase gained by using a different prediction strategy.

4 Text classification

4.1 Features

For author profiling, it has been shown that tf-idf weighted n-gram features, both in terms of characters and words, are very successful in inferring gender[9]. As such we have decided to use character 2- to 7-grams and word 1- to 3-grams with tf-idf weighting with sublinear term frequency scaling[12].

Word embeddings are a distributed representation for text that is perhaps one of the key breakthroughs for the impressive performance of deep learning methods on challenging natural language processing problems[13, 14]. They work in such a way that words with similar meaning get a similar representation in a lower dimensional space. These embeddings are trained on huge corpora of text to have the most context-specific information. For English and Arabic we used the pretrained fastText embeddings[15]. For Spanish the pretrained embeddings we used were trained on the Spanish Billion Word Corpus Embeddings[16].

4.2 Recurrent Neural Network

RNNs[4] are used to model sequences where the order is important. They have an internal memory that keeps track of the examples they have seen so far in the current sequence. Text is one of the clear use cases for RNNs[17] because of its sequential nature.

A challenge with using a recurrent neural networks is the vanishing gradient problem. In this problem long dependencies get lost over time. The problem was explored in depth in [18, 19] who found some fundamental reasons why it might be difficult to retain these dependencies. One solution to this problem is to use multiple gates as the atomic units within a recurrent neural network. Multiple versions of these gates exist, such as Long Short Term Memory-units (LSTM)[20] and Gated Recurrent Units (GRU)[21]. Chung et al. [22] note that both LSTM and GRU are superior over recurrent neural networks with traditional $tanh$ units. LSTMs are, in theory, better able to remember longer sequences than GRUs and outperform them in tasks requiring modeling long-distance relations. An advantage of GRUs over LSTMs is that they are computationally more efficient because they have fewer within the units. As noted in Section 2 the texts are small in length and as such we do not need the additional power of LSTMs and have decided to use GRUs.

Another way to solve the long term dependency problem is to use an attention mechanism. They were recently demonstrated to have success in a wide range of tasks[23, 24, 25, 26]. We use a modification of the mechanism proposed by Zhou et al. [27], in which we have not used the weighted sum but instead have taken the global maximum and the global average over the attention matrix and have concatenated the two.

Bidirectional RNNs[28, 29] are a combination of two seperate RNNs. The input sequence is fed in the normal order for one network, and in reverse order for the other. The outputs of the two networks are usually concatenated at each time step. This structure allows the networks to have both backward and forward information about the sequence at every time step. Human understanding of text works in the same way, we use the context of words to determine their meaning. In our work the seperate RNNs have the same configuration.

Recurrent neural networks can require millions of parameters to sufficiently model tasks. This high dimensional parameter space translates to a high chance of overfitting on the training data set. Because large networks are slow to use, creating an ensemble of many large networks is infeasible. One technique to reduce the overfitting is to add dropout[30, 31] to the network. We used different amounts of dropout in different places in the network. Between the embeddings and the recurrent layer of our network we

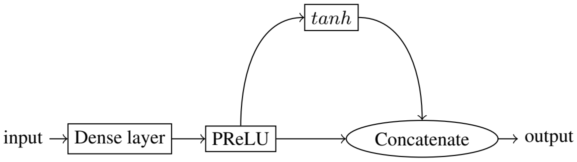

Figure 1. The dense block that we use instead of a single activation function.

use spatial dropout[32] instead of normal dropout. The benefit of this is that entire embedding channels can be dropped with a certain probability, which is better than removing random points in the embedding matrix.

We used 300-dimensional word embeddings as the input for our network. Spatial Dropout with a rate of 0.4 is applied to the word embeddings. We used a bidirectional GRU with 256 units for each direction. The GRU had a $tanh$ activation function, an output dropout-rate of 0.35 and an internal dropout-rate of 0.1. After the recurrent layer the attention mechanism was applied. Global average and max pooling were applied to this layer to get a single vector for every input text. The pooling operations are concatenated together as input to a dense network with a dropout rate of 0.5 between every dense block. A dense block consists of a fully connected layer with a PReLU[33] activation function. We also applied the $tanh$ activation function on the output of the PReLU and concatenated it together with the original, as seen in Figure 1. Three such dense blocks were used with respectively 256, 128 and 64 neurons. Because of the concatenation the output size of these blocks is twice the number of neurons. The final output was a single neuron with the sigmoid function. We optimize the model with the Adam[34] optimizer and as the loss function we chose binary cross entropy. The network architecture can be seen in Figure 2.

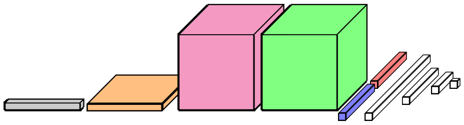

Figure 2. The layout for the recurrent neural networks we used for classifying the text. The gray layer is the input text. The orange layer is an embedding layer which replaces the words by their embedding. The magenta layer is the shape of the output of the bidirectional recurrent layer. The green layer is the attention mechanism. The blue layer is a global max pooling layer. The red layer is a global average pooling layer. The white layers are the dense blocks.

4.3 Ensemble

To do our final predictions on the texts we make use of an ensemble of two models. The ensemble is a combination of a traditional model and a deep-learning model. The deep-learning model (GRU) is described in the previous section.

The traditional model is a multinomial Naive Bayes[3] classifier (NB) using the character and word n-grams with tf-idf weighting on tweet level. Naive Bayes is a family of classification algorithms based on the assumption that every feature being classified is independent of any other feature given the class. The Naive Bayes classifier considers each word in a piece of text to contribute independently to the probability that the author is female (or male), regardless of any correlations between features. Although it is based on independence assumptions that often do not hold in the real world, Naive Bayes can often obtain surprisingly good results[35].

The ensemble uses a weighted average to combine the output of different models. The weights and models within the ensemble have the same architecture, regardless of language.

5 Image classification

After inspecting the images in the dataset we found that a lot of users post selfies. Our hypothesis is that if we could identify selfies, and detect the gender of a person on that selfie we could predict the gender of the author of the picture. If we do not find selfies for a user this pipeline will give a random prediction for the user.

Our image model is not our main approach to this shared task and as such we hope to improve our score in the combined task using it. Because of this it does not really make an impact on our results if a user does not post selfies.

For this approach we need a dataset consisting of selfies, and a dataset without selfies. For the selfies class we used the selfie dataset provided by Kaleyeh et al. [10]. In this research 46.836 selfies where collected and annotated with 36 different attributes. The focus of this research was to predict the popularity of a selfie. For the no-selfie class we used the MIRFLICKR dataset[11]. This dataset consists of 25.000 images from Flickr. The images are annotated with tags. We removed images containing the tags 'person', 'portrait' or 'selfie' resulting in 23.500 images.

In 2012 Krizhevsky et al. won the ImageNet competition with a CNN[36]. Since then they have been the default architecture to tackle computer vision problems. We have used a CNN to detect selfies and if it is we predict the gender of this selfie with a different CNN with the same architecture. The architecture we used is shown in Figure 3. There are 64 filters in every convolutional layer. The kernel-sizes are 3×3 and the max-pooling size is 2×2. In every layer except the last we used the ReLU[37] activation function. In the last dense layer it is a sigmoid. The selfie detection was trained for 20 epochs using the Adam[34] optimizer on 150px by 150px versions of the input images with a batch size of 256. We augmented the dataset by rescaling, zooming and shearing and horizontal flipping of the images. We got a 96 percent accuracy of correctly identifying a selfie on a validation set of our created dataset. We found that on a small sample over 80% of the users post images that get classified as selfies. For this model we got an accuracy of 86% on just selfies. The model does not perform well on images that are not selfies.

One caveat of this approach is that not every picture with a face posted is of the author themselves. However we hypothesize that more often than not women will post pictures of themselves or other women and likewise with men.

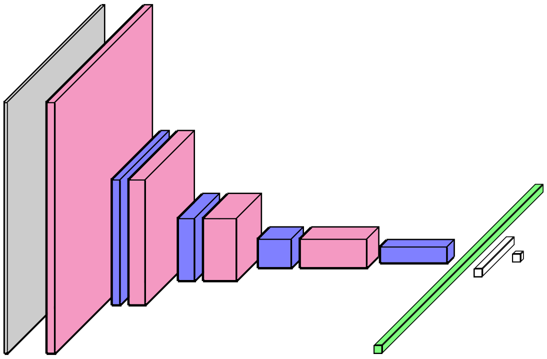

Figure 3. The layout for the convolutional neural networks used in the selfie identification and gender prediction. The gray layer is the input image. The magenta layers are convolutional layers. The blue layers are max-pooling layers. The green layer is a flattened version of the layer before it. The white layers are normal dense layers.

6 Combining models

To combine the text models and the image models we use a weighted average. Overall, the text models were vastly outperforming the image models, however the addition of the image models did improve the overall performance of the system.

We chose to keep a single configuration for all languages. The weighted mean between the text models is a 1:4 ratio in favor of the RNN model. This is also the case for the combination of the image model and the text models where the ratio is in favor of the text models.

7 Results

As in previous years with this shared task the models are compared using accuracy of correctly predicting the gender of an author. For every language the accuracy is calculated. Then, the accuracies are averaged to obtain a final score for our submission. The results in this section are evaluated on the PAN 2018 Author profiling evaluation set.

Table 2 shows the accuracy of our text models and ensemble. We achieve an accuracy of 76% for Arabic, 78.5% for English and 74.1% for Spanish on the evaluation set using the ensemble. There is a big difference in performance between Arabic and English, and Spanish. This might be because we have done additional preprocessing for Arabic and English. The GRU model outperforms the Naive Bayes model. The ensemble has a higher accuracy than the models separately for Spanish. For Arabic and English this is not the case, here the GRU model has the highest performance. To prevent overfitting on the very small test set we used for tweaking we did not alter our ensemble based on these results.

Using only the selfie model we get accuracies upwards of 62% of the different languages, as can be seen in Table 2. This low accuracy might be because not all users

	NB[†]	GRU[†]	NB+GRU	Image	Joint
Arabic	0.660	0.800	0.760	0.623	0.764
English	0.660	0.790	0.785	0.658	0.788
Spanish	0.640	0.720	0.741	0.623	0.743
Average	0.653	0.770	0.762	0.635	0.765

Table 2. The table shows the accuracy of 5 different models: 3 text models, 1 image model and 1 joint model per language. The text models are: NB, GRU and NB+GRU. We hypothesize that the big difference between languages for the text models comes from the preprocessing used varying amount of preprocessing that is done on the dataset. The image column shows the accuracy of the gender prediction pipeline using our selfie detection algorithm on the evaluation set. The joint column shows the results of the weighted combination of the NB+GRU and image models on the evaluation set. The joint model has a slight improvement over using just the text models.

post selfies so our model does not know what to predict. Another reason might be that the selfies in the shared task dataset are different from the ones in the selfie dataset. It might also be the case that the MIRFLICKR dataset might not be sufficiently diverse. The images in this dataset are all high quality photos, which is not necessarily the case for the images shared in the PAN '18 dataset. We note that the accuracy on the images shared by Spanish users is a lot higher than with the Arabic and English users. We postulate that Spanish users might post more selfies or images representative for gender. For this reason we could have chosen to make the weight of the image model higher in the combined model case. However, to prevent overfitting, we have not done this.

The addition of the image models to the text models did give a very small improvement to the accuracy of our models (0.3%). This is because there is a big difference between the performance of the two approaches. If the performance of our image models would be on the same level as our text models we would see a significant improvement by using an ensemble of the two.

8 Conclusion

In this paper we have used a combination of text models and image models to create gender predictions for three different languages. We have done the predictions on individual tweets and images and then used multiple strategies to combine these predictions to create a single prediction on user level. We have also chosen to keep a single configuration of the system across the languages.

As such our performance on the individual languages is not as high as it could have been, had we optimized every combination of models for the different regions.

[†] The results of the NB and GRU models are obtained by evaluating the models on a small test set of 100 users as it was not possible to run the models on the evaluation set used. As such they might not be entirely representable for the performance of our models. We show these results for completeness.

An ensemble of a RNN and a bag of words model did improve performance on the English language, with respect to just using the RNN, but it does not improve on the other languages.

On the evaluation set, we got accuracy scores between 62.3% and 78.8% depending on language and whether we used models that classify based on text or on images. On our small test set our non-ensemble models showed an improved performance, however the test set only contained 100 users and as such were not be representable for the distributions shown in the evaluation set.

To conclude: we successfully defined an ensemble of deep-learning and traditional models capable of good performance.

References

1. Gómez-Adorno, H., Markov, I., Sidorov, G., Posadas-Durán, J.P., Sanchez-Perez, M.A., Chanona-Hernandez, L.: Improving feature representation based on a neural network for author profiling in social media texts. Computational intelligence and neuroscience **2016** (2016) 2
2. Stamatatos, E., Rangel, F., Tschuggnall, M., Kestemont, M., Rosso, P., Stein, B., Potthast, M.: Overview of PAN-2018: Author Identification, Author Profiling, and Author Obfuscation. In Bellot, P., Trabelsi, C., Mothe, J., Murtagh, F., Nie, J., Soulier, L., Sanjuan, E., Cappellato, L., Ferro, N., eds.: Experimental IR Meets Multilinguality, Multimodality, and Interaction. 9th International Conference of the CLEF Initiative (CLEF 18), Berlin Heidelberg New York, Springer (September 2018)
3. Hand, D.J., Yu, K.: Idiot's bayes — not so stupid after all? International Statistical Review **69**(3) 385–398
4. Lipton, Z.C., Berkowitz, J., Elkan, C.: A critical review of recurrent neural networks for sequence learning. (2015)
5. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553) (2015) 436
6. Rangel, F., Rosso, P., Montes-y-Gómez, M., Potthast, M., Stein, B.: Overview of the 6th Author Profiling Task at PAN 2018: Multimodal Gender Identification in Twitter. In Cappellato, L., Ferro, N., Nie, J.Y., Soulier, L., eds.: Working Notes Papers of the CLEF 2018 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (September 2018)
7. Basile, A., Dwyer, G., Medvedeva, M., Rawee, J., Haagsma, H., Nissim, M.: N-GrAM: New Groningen Author-profiling Model. (July 2017)
8. Rangel, F., Rosso, P., Verhoeven, B., Daelemans, W., Potthast, M., Stein, B.: Overview of the 4th author profiling task at pan 2016: Cross-genre evaluations. In: Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings, Évora, Portugal, CLEF and CEUR-WS.org, CLEF and CEUR-WS.org (2016/09 2016)
9. Rangel, F., Rosso, P., Potthast, M., Stein, B.: Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter. Working Notes Papers of the CLEF (2017)
10. Kalayeh, M.M., Seifu, M., LaLanne, W., Shah, M.: How to take a good selfie? In: Proceedings of the 23rd ACM International Conference on Multimedia. MM '15, New York, NY, USA, ACM (2015) 923–926
11. Huiskes, M.J., Lew, M.S.: The mir flickr retrieval evaluation. In: MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval, New York, NY, USA, ACM (2008)
12. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge, UK (2008)

13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. (2013)
14. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). (2014)
15. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. CoRR **abs/1802.06893** (2018)
16. Cardellino, C.: Spanish Billion Words Corpus and Embeddings (March 2016)
17. Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: Eleventh Annual Conference of the International Speech Communication Association. (2010)
18. Hochreiter, S.: Untersuchungen zu dynamischen neuronalen netzen. Diploma, Technische Universität München **91** (1991) 1
19. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies (2001)
20. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8) (1997) 1735–1780
21. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv:1406.1078 [cs, stat] (June 2014)
22. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
23. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
24. Hermann, K.M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P.: Teaching machines to read and comprehend. In: Advances in Neural Information Processing Systems. (2015) 1693–1701
25. Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y.: Attention-based models for speech recognition. In: Advances in neural information processing systems. (2015) 577–585
26. Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., Jin, Z.: Classifying relations via long short term memory networks along shortest dependency paths. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. (2015) 1785–1794
27. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Volume 2. (2016) 207–212
28. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing **45**(11) (1997) 2673–2681
29. Zhang, S., Zheng, D., Hu, X., Yang, M.: Bidirectional long short-term memory networks for relation classification. In: Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation. (2015) 73–78
30. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**(1) (2014) 1929–1958
31. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization. arXiv preprint arXiv:1409.2329 (2014)
32. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient Object Localization Using Convolutional Networks. arXiv:1411.4280 [cs] (November 2014)

33. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. (2015) 1026–1034
34. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization
35. Zhang, H.: The optimality of naive bayes. (2004)
36. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
37. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853 (2015)