COMP5331 Project

Exploring Current Models on Visual Information for Fake News Detection

Group 14 **Type: Implementation**

CHENG, Tat Fan
GUPTA, Ayush
LIU, Songyu
LUNG, Kun Hung
YAN, Chiu Wai

Presented by
CHENG Tat Fan

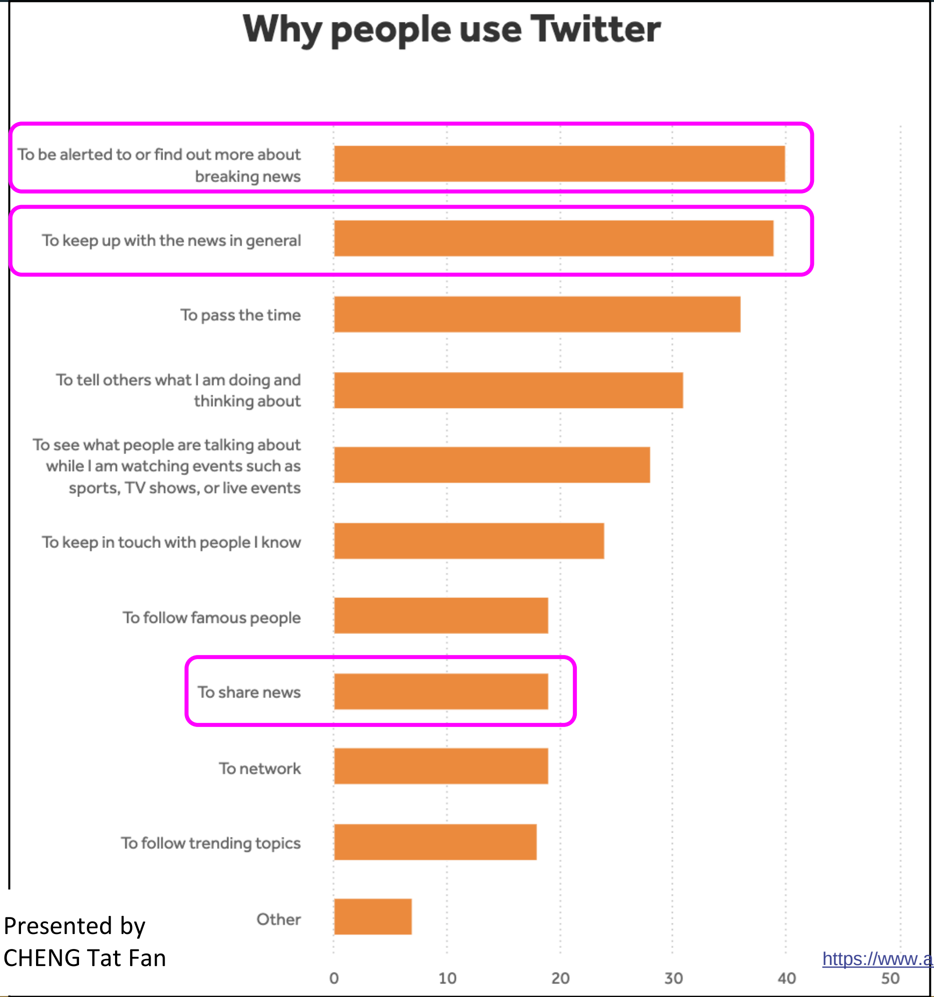
Overview

- Introduction & Dataset
  - Background, Previous work, Objectives, Datasets
- Implementation & Results
  - Baseline models, the full MVNN model and its child models
- Conclusion

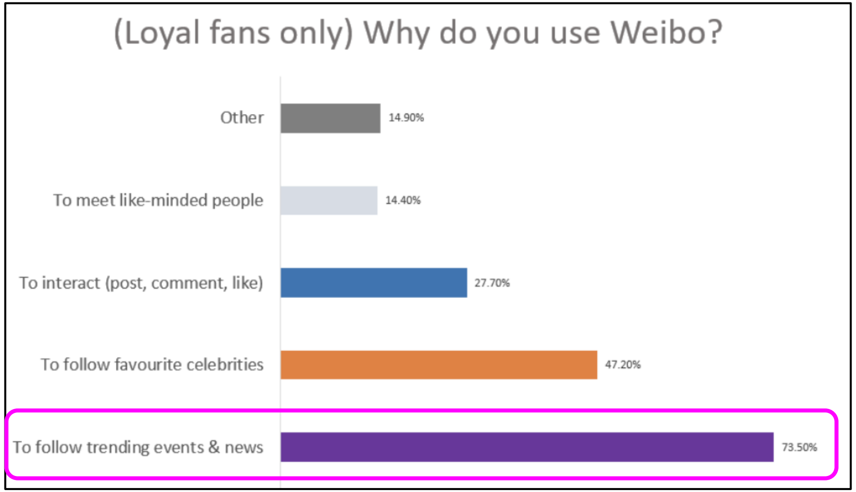
Background

- Exponential growth of online news media
  - More accessible, interactive and vivid news
  - Downplay the role of traditional news reporting
- Apart from these, networking platforms become convenient places for sharing news.

In the old times: newspapers, TV, radio, etc.



Twitter and Weibo are popular platforms for seeking & sharing news, **but also a breeding ground for fake news & rumor**



Presented by
CHENG Tat Fan

https://www.americanpressinstitute.org/publications/reports/survey-research/how-people-use-twitter-news/single-page/
https://www.sekkeistudio.com/blog/2016-weibo-user-research-report/

Features of fake news

- Tempered images
- Misleading images
- Real images but with misleading text

(Qi et al. 2019)

Obama makes shocking remarks on terrorism. **(Scraped Fake News)**

Obama found to support terrorism. **(Scraped Fake News)**

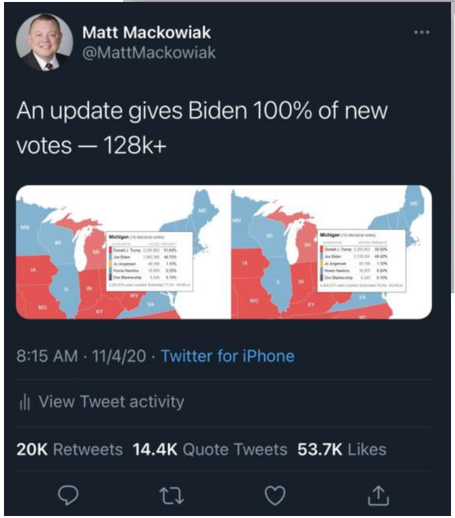
Obama gives a speech on terrorism. **(Additional Scraped Real News)**

Obama makes shocking remarks on terrorism. Obama found to support terrorism. **(Intelligently Generated Fake News)**

(Jindal et al. 2020)

A real-world example

- About the Michigan voting map in US Election 2020
    - A sudden increase of 128k+ votes for Joe Biden, but none for President Trump.
    - It turns out to be an vote count updated after correcting a data entry error.
    - But the image was used to advocate a vote fraud in the election
    - It first appeared on a message board called “8kun” → Twitter and was retweeted by a conservative commentator.
    - The post went viral even after the original tweet was deleted.



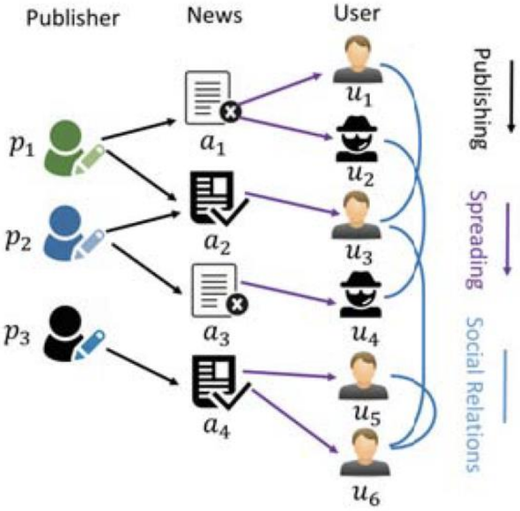
Presented by
CHENG Tat Fan

Sources: https://www.bbc.com/news/election-us-2020-54838684
https://www.bbc.com/news/election-us-2020-54811410

Previous work

- Natural language processing (kwon et al. 2013)
  - Eye-catching or negative words
  - Hard to mine the discriminative textual features (Jindal et al. 2020)
- Relationship between the publisher, news and users (Shu et al. 2018)

(Jindal et al. 2020)

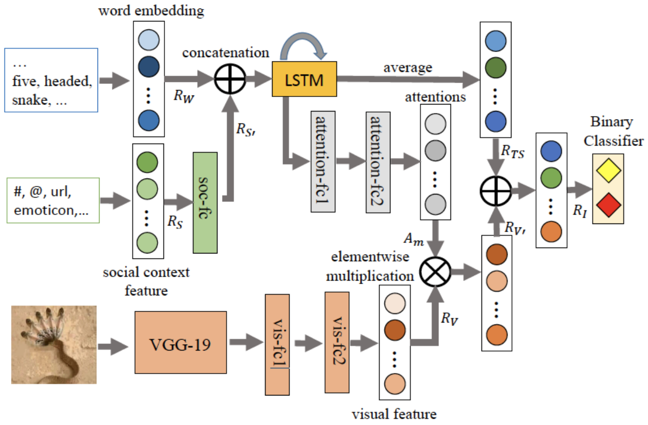


(Shu et al. 2018)

Presented by
CHENG Tat Fan

Previous work

- RNN with an attention mechanism (att-RNN) (Jin et al. 2017)
  - Fusion of the image, textual and social context features extracted by VGG and LSTM



(Jin et al. 2017)

Table 4: The results of component analysis on two datasets

Method	Weibo Accuracy	Weibo Rumor F_1	Weibo Non-rumor F_1	Twitter Accuracy	Twitter Rumor F_1	Twitter Non-rumor F_1
att-RNN	**0.788**	**0.764**	**0.807**	**0.682**	**0.689**	**0.676**
w/o attention	0.745	0.71	0.773	0.668	0.68	0.655
w/o social	0.772	0.742	0.795	0.664	0.676	0.651
w/o social+attention	0.736	0.706	0.76	0.631	0.611	0.65
w/o image	0.743	0.708	0.771	0.625	0.642	0.605
w/o image+social	0.721	0.683	0.752	0.613	0.693	0.474

Presented by
CHENG Tat Fan

Previous work

- Multi-domain Visual Neural Network (Qi et al. 2019)
  - CNN & GRU to mine image features in
    - frequency domain (re-compression/low image quality)
    - pixel domain (emotional/eye-catching)
  - **84.6%** accuracy (on the same datasets as Jin et al. 2017)
  - **This MVNN model is implemented in our project.**

(Qi et al. 2019)

TABLE I: The performance comparison of fake news detection in single visual modality.

Method	Accuracy	Precision	Recall	F1
FF+LR	0.650	0.612	0.579	0.595
Pre-trained VGG	0.721	0.669	0.738	0.702
Fine-tuned VGG	0.754	0.74	0.689	0.714
ConvAE	0.734	0.685	0.744	0.713
MVNN	**0.846**	**0.809**	**0.857**	**0.832**

Introduction - Objectives

Use MVNN to extract visual features from different domains and merge them using attention

Compare the performance of MVNN with other baselines

Introduction - Objectives

Test the performance of the MVNN model with parts of it removed

Tune hyper parameters

Datasets

- Weibo Dataset (This is the one we used for training)
- Twitter Dataset (not used)

The Twitter Dataset

Roughly 350 Images.

Too small for Deep learning

MM17-WeiboRumorSet

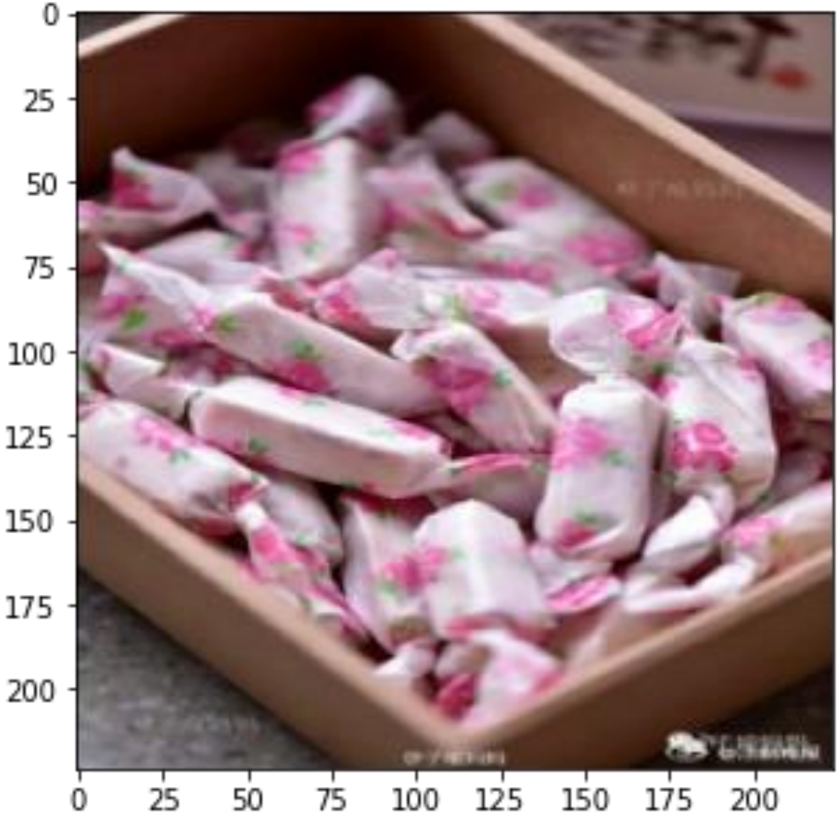
[nonrumor_images]: 5318 images.

[rumor_images]: 7954 images.

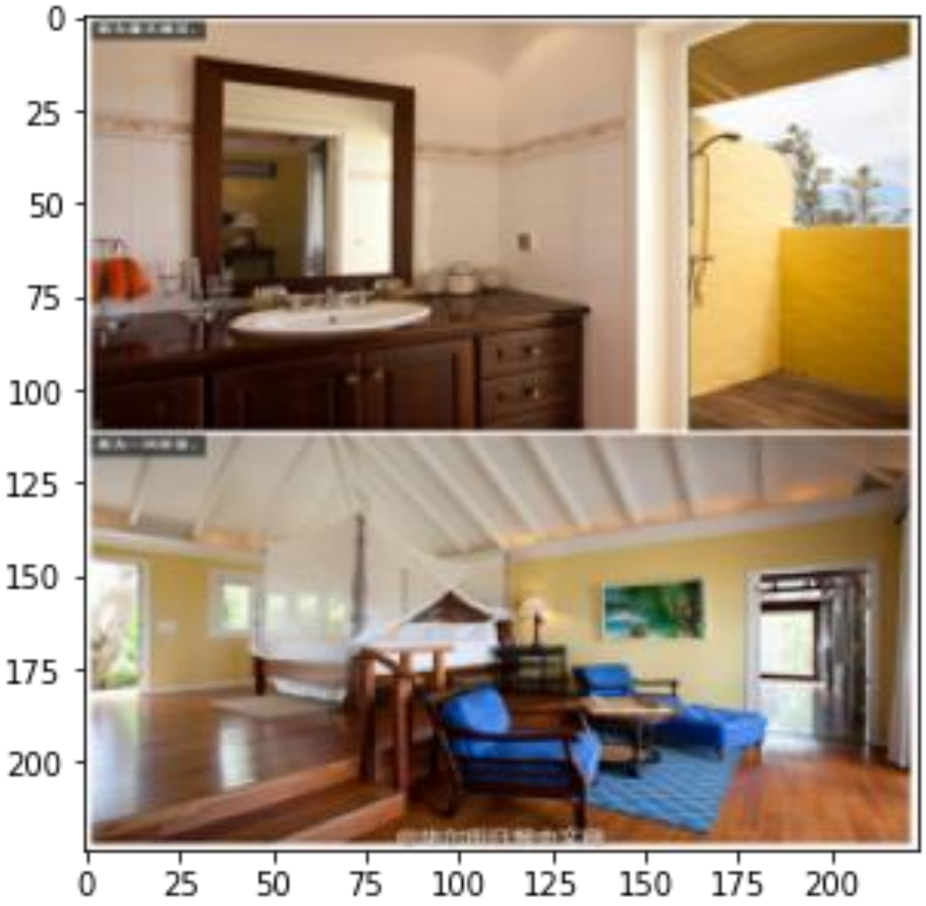
Total no. of images: 13272

Large enough for Deep learning.

Non rumor



Presented by
LIU Songyu


0
25
50
75
100
125
150
175
200
0
25
50
75
100
125
150
175
200

Rumor


@肿瘤内科专家
weibo.com/oyxn


@-完美动物-
weibo.com/u/1013814961


著名好莱坞坞影星尼古拉斯.凯奇于
2013年1月17日和家人朋友去滑雪时
发生意外去世年仅49岁.据目击者称
尼古拉斯滑雪过程中滑雪板失去控制
无法停下高速撞到树上当场死亡.
@Will-pan215
weibo.com/u/1012469064

Methodology - Implementation

- Python, Google Colab GPU
- Baseline Models
    - VGG-16 VGG-19
- Simple Sequential model of CNN
- MVNN based models from paper [1]
    - Frequency based - Attention
    - Pixel based - Full MVNN

Baseline Models

- VGG-16
- VGG-19
- Sequential model of CNN

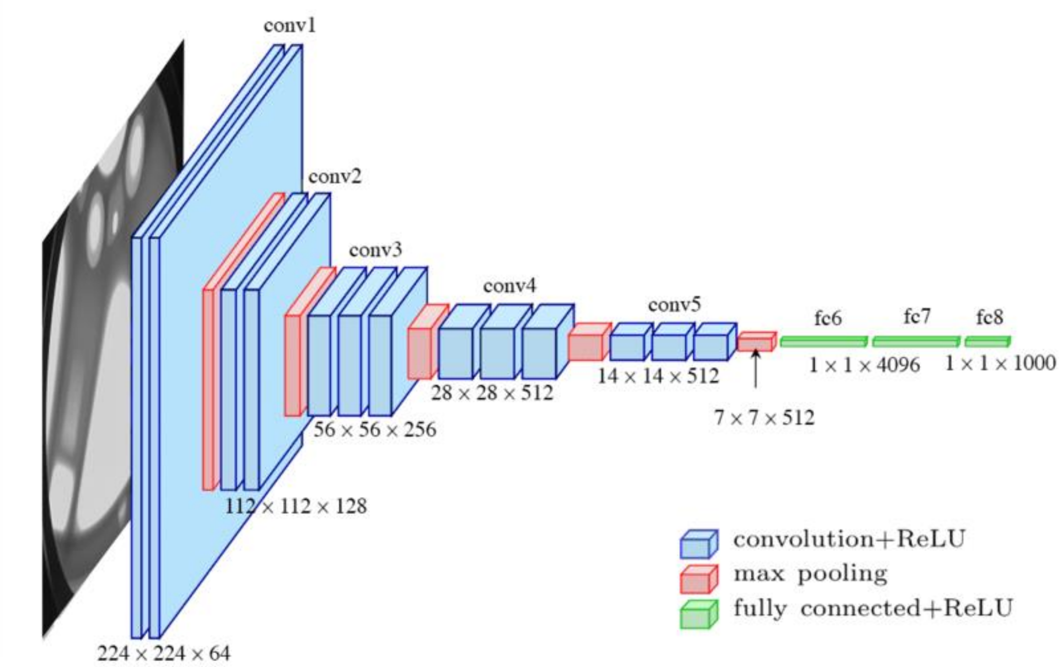
Presented by
GUPTA Ayush

Baseline Models

- VGG-16 & VGG-19 - popular models
- Pre-trained on ImageNet
- Fine tuned on Weibo data set

Presented by
GUPTA Ayush

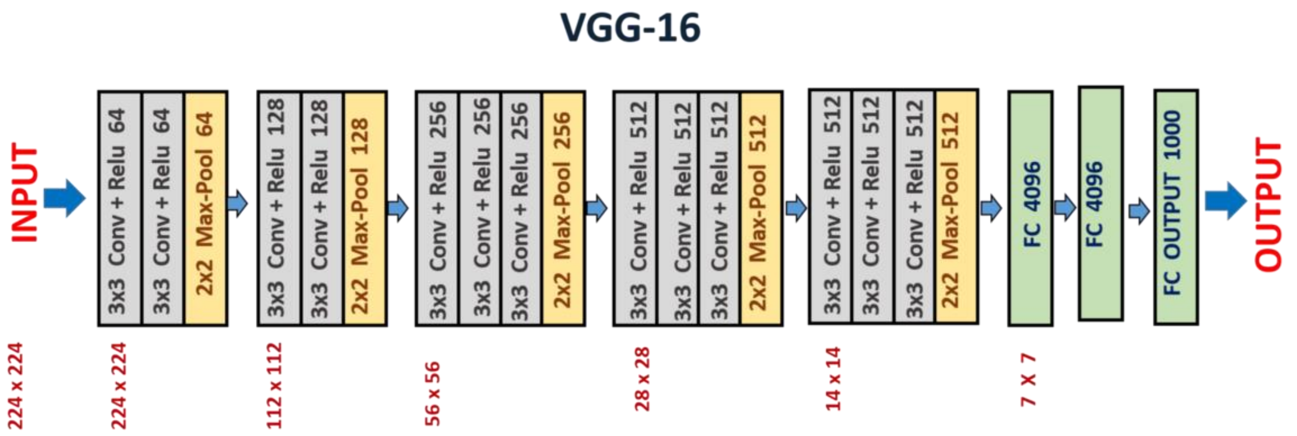
Baseline Model - VGG-16



Presented by
GUPTA Ayush

Image from Researchgate.net

Baseline Model - VGG-16



Presented by
GUPTA Ayush

Image from Researchgate.net

Simple Sequential model of CNN

- Simplified version of VGG-16
- 2D CNN, ReLu activation,
- Batch Normalization, MaxPooling2D, Dropout layers
- Fully Connected layer

Simple Sequential model of CNN

- **Architecture:**
- Conv2D, BatchNorm, MaxPool2D, Dropout, Conv2D, BatchNorm, MaxPool2D, Dropout
- 126*126*32, 126*126*32, 63*63*32, 63*63*32, 61*61*64, 61*61*64, 30*30*64, 30*30*64

- Fully Connected (FC) - Flatten, Dense, BatchNorm, Dropout, Dense
- No.: 57600, 512, 512, 512, 512

- No of layers of CNN: 1,2,3,4. Hyperparameter tuning.
- 2 found as best.

Presented by
GUPTA Ayush

Results - Baseline Model

Simple Sequential model of CNN:

Num of CNN layers	Precision	Recall	f1-score	Accuracy	Cohen Kappa
1	61%, 72%	56%,76%	59%,74%	66%	0.332
2	68%, 77%	67%,78%	68%,77%	73%	0.451
3	58%, 78%	71%,66%	64%,72%	70%	0.357
4	58%, 78%	71%,66%	64%,72%	68%	0.356

VGG:

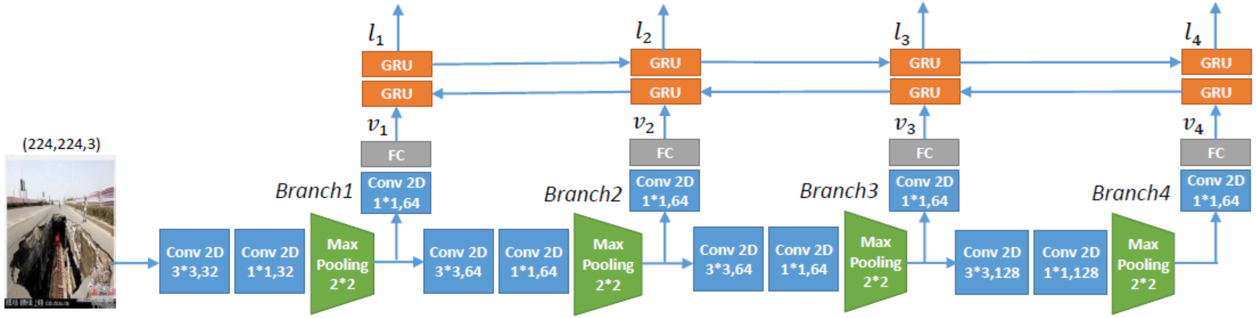
Type of model	Precision	Recall	f1-score	Accuracy	Cohen Kappa
VGG-16	69%, 73%	57%,82%	62%,78%	72%	0.40
VGG-19	59%, 79%	76%,64%	66%,71%	69%	0.38

MVNN - Overview

- Implemented based on Paper [1]
- Multi-domain Visual Neural Network (MVNN) framework
- Frequency - DCT (Discrete Cosine Transform) - CNN
- Pixel - CNN-RNN-GRU
- Attention - Combine the above, Train
- Full model

MVNN - Pixel Domain

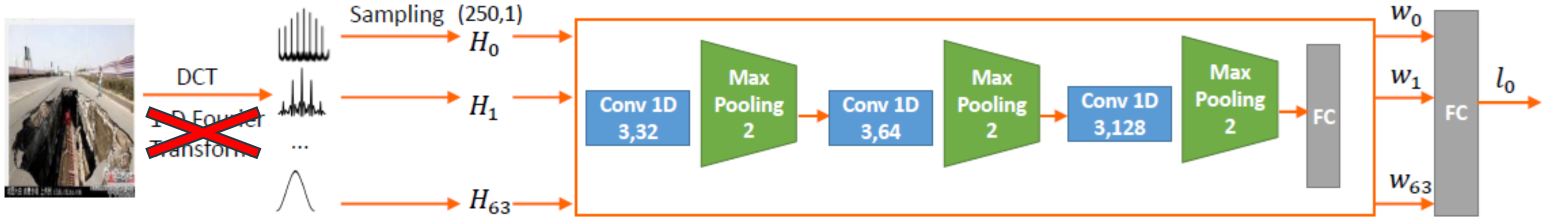
- 4-branch CNN-RNN
- Extract feature at each semantic level
- Bi-GRU (Bidirectional GRU)
    - Input: 4 32-vectors
    - Output 4 64-vectors $L = \{l_1, l_2, l_3, l_4\}$, where $l_t = [GRU_f(v_t), GRU_b(v_t)]$



MVNN - Frequency Domain

Idea: JPEG images produce traces during re-compression, which can be detected by examining its **8x8 DCT Blocks** [5]

- Preprocess data:
  - Reshape into (128, 128, 3), in YCbCr channels
  - 8x8 DCT (Discrete Cosine Transform)
- Backbone: 1D Convolutional Neural Network



Presented by
YAN Chiu Wai

MVNN - Attention

Idea: Let the model decide which feature should contribute more to the final result depending on the situation

- Previous outputs: 5 64-vectors of features
  - 4 from pixel domain, 1 from freq. domain
- Output u, a single 64-vector
- FC layer with softmax activation outputting 2 logits: P(FAKE) and P(REAL)
- Loss function: cross-entropy loss

$$L = -\sum \left[y \log p + (1-y) \log (1-p) \right]$$

$$\mathcal{F}(l_i) = v^T \tanh(W_f l_i + b_f), i \in [0, 4]$$

$$\alpha_i = \frac{\exp\left(\mathcal{F}\left(l_i\right)\right)}{\sum_i \exp\left(\mathcal{F}\left(l_i\right)\right)}$$

$$u = \sum_i \alpha_i l_i$$

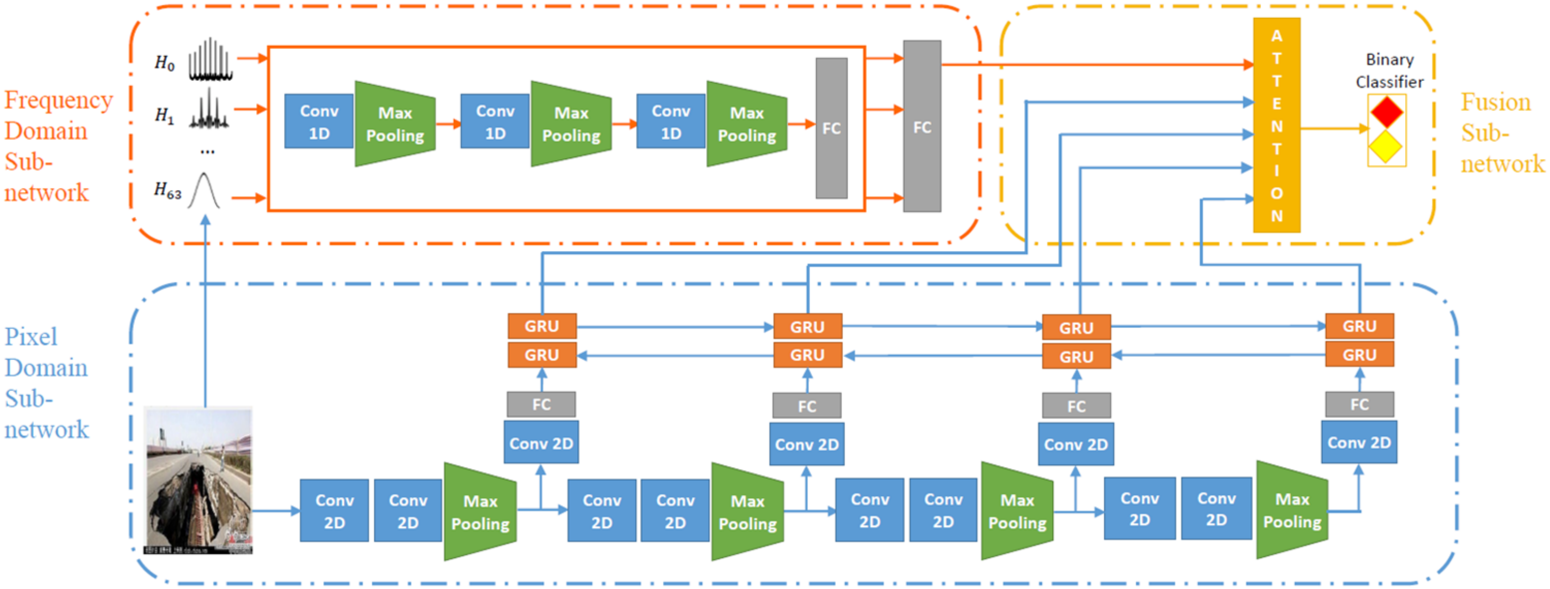
v is a weight vector (parameter to learn)

F is the score function evaluating significance of a feature vector

a_i is the final weight assigned to each vector

u is the output 64-vector

MVNN - Full



Presented by
YAN Chiu Wai

Results - Metric

- Precision
- Recall
- f1-score
- Accuracy

Results - Metric

- ## Cohen Kappa
  - measure inter-rater reliability for qualitative items
  - consensus exists in the ratings given by various judges

< 0.00	Poor
0.00 - 0.20	Slight
0.20 - 0.40	Fair
0.40 - 0.60	Moderate
0.60 - 0.80	Substantial
0.80 - 1.00	Almost perfect

Results - Baseline Model

Simple Sequential model of CNN:

Num of CNN layers	Precision	Recall	f1-score	Accuracy	Cohen Kappa
1	61%, 72%	56%,76%	59%,74%	66%	0.332
2	68%, 77%	67%,78%	68%,77%	73%	0.451
3	58%, 78%	71%,66%	64%,72%	70%	0.357
4	58%, 78%	71%,66%	64%,72%	68%	0.356

VGG:

Type of model	Precision	Recall	f1-score	Accuracy	Cohen Kappa
VGG-16	69%, 73%	57%,82%	62%,78%	72%	0.40
VGG-19	59%, 79%	76%,64%	66%,71%	69%	0.38

Results - MVNN Final

Precision	Recall	f1-score	Accuracy	Cohen Kappa
76%, 87%	80%,84%	78%,85%	82.2%	0.631

TABLE II: Ablation study of MVNN.

Method	Accuracy	Precision	Recall	F1
MVNN	**0.846**	**0.809**	**0.857**	**0.832**
w/o frequency domain	0.794	0.792	0.728	0.758
w/o pixel domain	0.737	0.698	0.717	0.708
w/o attention	0.827	0.778	0.853	0.814
w/o Bi-GRU	0.828	0.772	0.841	0.805
w/o branches	0.803	0.752	0.830	0.789

Results - MVNN Variations

w/o pixel domain

Precision	Recall	f1-score	Accuracy	Cohen Kappa
65%, 76%	61%,79%	63%,77%	72%	0.404

w/o frequency domain

Precision	Recall	f1-score	Accuracy	Cohen Kappa
79%, 83%	73%,87%	76%,85%	81.2%	0.603

w/o attention

Precision	Recall	f1-score	Accuracy	Cohen Kappa
75%, 86%	79%,83%	77%,85%	81.8%	0.620

w/o GRU

Precision	Recall	f1-score	Accuracy	Cohen Kappa
76%, 85%	78%,83%	77%,84%	80.9%	0.606

w/o branches

Precision	Recall	f1-score	Accuracy	Cohen Kappa
76%, 86%	79%,83%	78%,84%	81.6%	0.620


Results
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
VGG-16
VGG-19
SimpleCNN
MVNN
...w/o pixel
...w/o freq
...w/o atten
...w/o GRU
...w/o branches
Precision
Recall
f1-score
Accuracy
CohenKappa

Conclusion

- Baseline Models
- MVNN
  - High accuracy (>80%)
  - Pixel domain is strongly correlated to MVNN
- Compare the result statistics

Conclusion

1. Stronger units: The Use of GRU can be replaced by more sophisticated architecture like LSTM or even TRANSFORMER
2. Dataset: Also including the TEXT part since fake news detection depends a lot in the actual text describing the image
3. More preprocessing: haven't test data augmentation due to the assumption on JPEG image (which is found to be not the most helpful)
4. Transfer learning: use existing convolution parameters instead of training from scratch and using randomly initialized weights

Thank you! :-)

Any questions for us?

References

1. **P. Qi, J. Cao, T. Yang, J. Guo, and J. Li. Exploiting multi-domain visual information for fake news detection. In 2019 IEEE International Conference on Data Mining (ICDM), pages 518–527, 2019.**

1. Chen, Yi-Lei and Chiou-Ting Hsu. Detecting recompression of jpeg images via periodicity analysis of compression artifacts for tampering detection. IEEE Transactions on Information Forensics and Security, 6:396–406, 062011

1. Jin, Zhiwei, Juan Cao, Han Guo, Yongdong Zhang, and Jiebo Luo. Mul-timodal fusion with recurrent neural networks for rumor detection on microblogs. In Proceedings of the 25th ACM international conference on Multimedia, pages 795–816. ACM, 2017.

1. Jindal, Sarthak, Raghav Sood, R. Singh, Mayank Vatsa, and Tanmoy Chakraborty. Newsbag: A benchmark multimodal dataset for fake news detection. In SafeAI@AAAI, 2020.

1. Kwon, Sejeong, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. Prominent features of rumor propagation in online social media. In 2013 IEEE 13th International Conference on Data Mining, pages 1103–1108. IEEE, 2013.

1. Shu, Kai, Suhang Wang, and Huan Liu. 2019. Beyond News Contents: The Role of Social Context for Fake News Detection. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19). Association for Computing Machinery, New York, NY, USA, 312–320. DOI:https://doi.org/10.1145/3289600.3290994

Additional information - Appendix

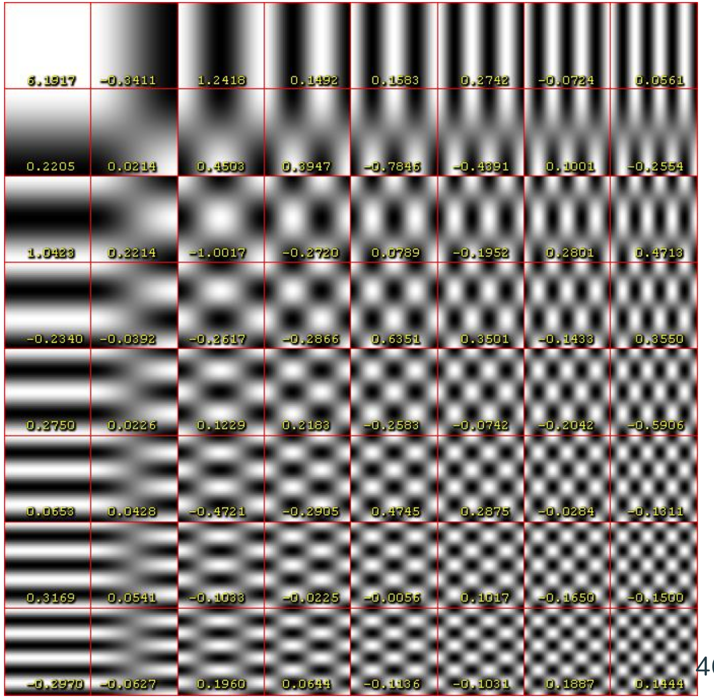
- Possible questions
  - Did we implement paper’s model successfully?
  - How good is our implementation w.r.t. Paper’s?

VGG-16

- https://medium.com/towards-artificial-intelligence/the-architecture-and-implementation-of-vgg-16-b050e5a5920b
- **VGG19** is a variant of **VGG model** which in short consists of **19** layers (16 convolution layers, 3 Fully connected layer, 5 MaxPool layers and 1 SoftMax layer).

DCT in Frequency Domain

- A discrete cosine transform expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. The DCT, first proposed by Nasir Ahmed in 1972, is a widely used transformation technique in signal processing and data compression.
- Source: Wikipedia



Hyper-parameters used in MVNN training

Images are resized to

- ❑ 128*128 for the **frequency subnetwork**
- ❑ 224*244 for the **pixel subnetwork**

80% training and 20% validation (Weibo dataset)

Learning rate = 0.0001 (best rate for converging the loss)

Batch size = 32

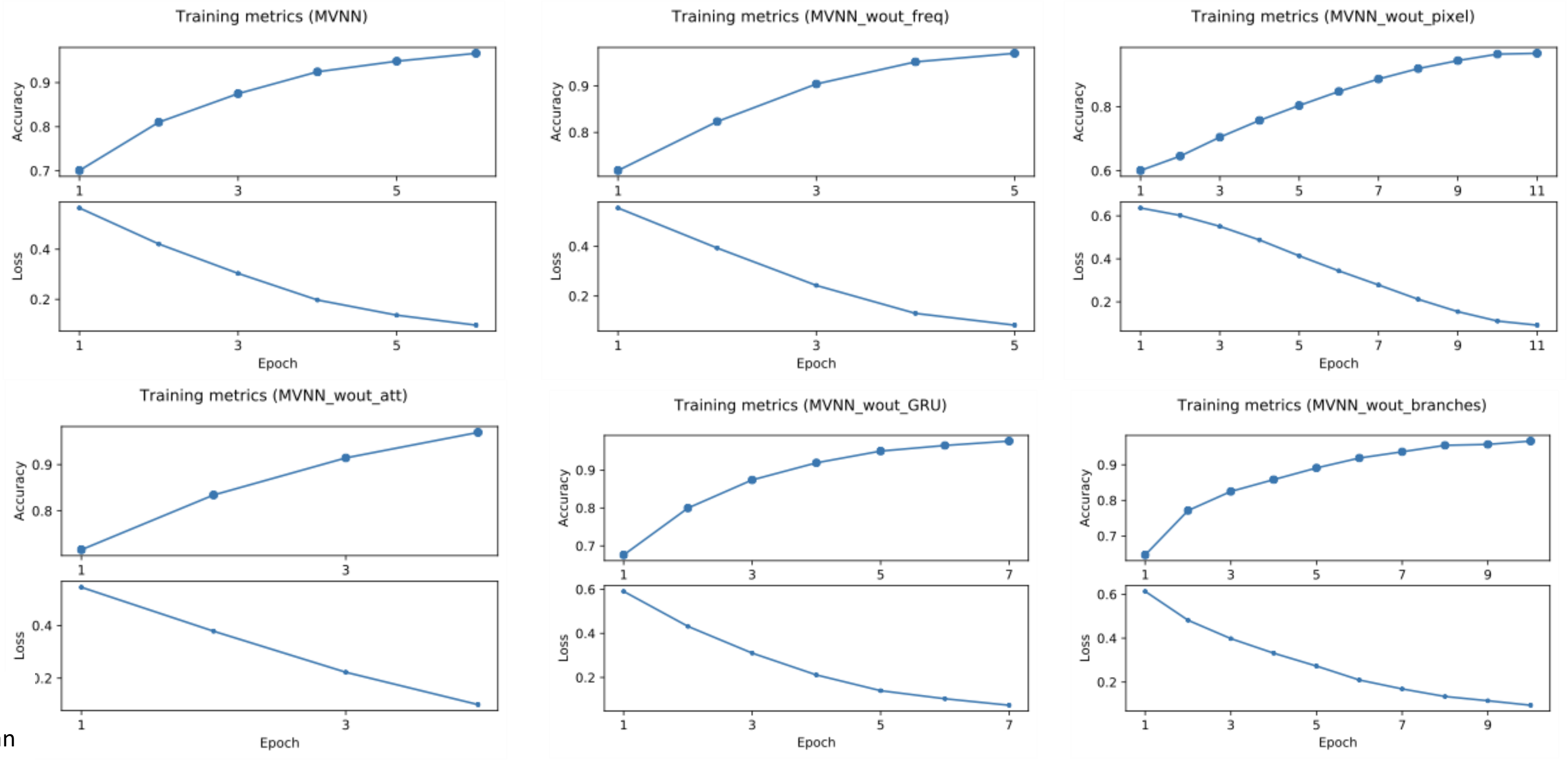
Cross-entropy Loss with the Adam optimizer (Qi et al., 2019)

Dropout probability = 0.5 (Qi et al., 2019)

No. of branches in pixel = 4 (suggested by Qi et al., 2019)

Training time

Training time per epoch (on GRU):
4-5 mins for all MVNN models



Prepared by
CHENG Tat Fan

Using trained MVNN on a completely new set

- **Twitter image dataset (360 images)**
- **The performance is not good**
  - Two completely different *cultures*
  - Different features of fake news
    - Weibo → tempered/exaggerated images
    - Twitter → misleading texts with real/misleading images
  - Overfitting of the model
    - Although we have an early stopper for training (when loss < 0.1)
    - Dropout layers (p = 0.5)

```
Accuracy: 0.4772727272727273
Balanced Accuracy: 0.4755707688830045
Confusion Matrix:
 [[ 65 108]
 [ 76 103]]
Cohen Kappa Score: -0.049010398108257025
Classification Report:
              precision    recall  f1-score   support

   non-rumor       0.46      0.38      0.41       173
       rumor       0.49      0.58      0.53       179

    accuracy                           0.48       352
   macro avg       0.47      0.48      0.47       352
weighted avg       0.47      0.48      0.47       352
```

Training full MVNN on a completely new set

- Twitter image dataset (360 images)
- The performance is decent

```
===== Start Validating ... =====
[Test] 2 / 2 batches tested
Accuracy: 0.703125
Balanced Accuracy: 0.6990196078431372
Confusion Matrix:
 [[26  8]
 [11 19]]
Cohen Kappa Score: 0.4003944773175543
Classification Report:
              precision    recall  f1-score   support

   non-rumor       0.70      0.76      0.73        34
       rumor       0.70      0.63      0.67        30

    accuracy                           0.70        64
   macro avg       0.70      0.70      0.70        64
weighted avg       0.70      0.70      0.70        64
```

Prepared by
Liu songyu