Using Eye Gaze Data to Predict Image Manipulation by Bi-Directional MLP, CNN and LSTM Models

Haoyang Zhang
Research School of Computer Science,
Australian National University, Canberra Australia
u6921074@anu.edu.au

Abstract. In this paper I followed the research of eye gaze investigation of human perceptions of manipulated and unmanipulated digital images and consider the leaving question: can subconsciously eye gaze data be a way to detect manipulated image instead of artificial cognition? The eye gaze data in this paper is the participants' eye fixation position-duration data on a group of same size manipulated/no manipulated images with the start and end timestamp. This project discussed 3 forms of usage of eye gaze data and used them to predict image manipulation in related 3 models: Bi-Directional MLP, CNN and LSTM. I did the k-fold and calculate the accuracy of these models and eventually found the limitation of the model performance on this eye gaze dataset.

Keywords: eye gaze, manipulated image, Bi-directional neuron network, LSTM, CNN

1 Introduction

"Photography lost its innocence many years ago. In as early as the 1860s, photographs were already being manipulated, only a few decades after Niepce created the first photograph in 1814." [1] Image manipulation can be used for different purposes and some of them can be unethical. The image manipulation is highly used to deceive the public and the demand of manipulation detection is increasing in 21th because of the fashion of social media which are concerned full of unverified information and the potential places of image deception crime. However, Shao [2] said nowadays everyone who has a computer and access to internet can technically use "Deep Fake" to make manipulated image and these "deep fake" images are hard to detect. In the future, humans will be easily tricked by Ai's manipulation. It is a challenge for the researches to find high performance methods to against the development of deep fake techniques.
Though it is difficult for humans to distinguish the manipulated images, humans still can "find" the difference by their eyes. In a research of Florian's [3], they found that the human eye gaze data show difference to the different types of distorted scenes. It proved that the informative cues can be extracted from eye gaze data and the sensitive human eye gaze is helpful in the manipulated scenes detection.
Additionally, in Conklin, K 's book [4], it said that "Eye-tracking is quickly becoming a valuable tool in applied linguistics research as it provides a 'real-time', direct measure of cognitive processing effort." I assume that the message contained in the eye gaze may help us predict the property of the observed image as well as it did in linguistic genre.
For the purpose of proving whether it can be useful or not. I was working with the investigation of human perceptions of manipulated imaged [5]. The two groups of participants were asked to tell the manipulated images from unmanipulated images, the eye gaze of participants was recorded using two Facelab 5.0.2 infra-red cameras.
The researchers set target manipulated rectangle regions on the images and records the number and duration of fixations inside and outside the target regions. They also record the timestamp of the fixations. The participants would answer questions about the images such as whether the image was manipulated and could you tell which part of things were manipulated after informing about the technique of image processing.
Eventually, the research found that the accuracy of recognize a unmanipulated image is higher than recognizing a manipulated image. However, the results showed that participants could not have a high accuracy of recognizing a image is manipulated or not especially could not recognize which part is manipulated. But the research also shows that the numbers and durations of fixations inside the target area of a manipulated image is correlation to the correctness of the vote of the participants.

I chose to predict the manipulation by 3 formats of input of eye gaze data:
model1. Using the statistic of pre-defined rectangle regions in the research to predict (Caldwell_ImageManipulation-EyeGaze_DataSetCombined_edit.xlsx)
model 2&3: not using a pre=defined regions but using the single points locations and values.
model2. Constructing the spatial matrix of [x_pos, y_pos, duration] which is a sparse matrix, the other points which are not appeared in the Caldwell_Manip_Images_10-14_TimeSeries.csv are padding to 0. It has the same shape with the images in the research (1360 x 1024)
model3. The input is a series of points with the order of recorded timestamp so that it is a sentence, each word is [x_pos, y_pos, duration], all words are in the order of timestamp.

2 Method

2.1 Dataset preprocessing

1. Some data only appear in the Caldwell_ImageManipulation-EyeGaze_DataSetCombined.xlsx but not in the Caldwell_Manip_Images_10-14_TimeSeries.csv. These are the inconsistent data so I dropped them.
2. The input of the model LSTM need to be in the time order. So, I sorted the TimeSeries in the order of increasing 'Start Time' and saved as TimeseriesInOrder.csv.
3. "At an individual level, the characteristics of each image (semantics and content) yielded varying outcomes aligned with the nature of the image itself" [5] The model needs to be image-centric so that I should only use the data of the same image to train models 2&3 but use image as an attribute for model 1. (Also use one hot encode to transform the categorical attribute image to 5 binary attributes) I grouped the data and split the files into 5 different images'. In this study, I only used the data of image 10. (Combined_image0.csv, Timeseries_image10.csv)
4. The original data of Timeseries is in the form of sparse data, I need to convert it from sparse to density data
5. Each sentence (image points sequence for LSTM) has different number of words (points) so that in each batch, I need to make all sentence in the same length. (By padding 0s) Also the input tensor needs to be in the form [batch, seq, feature dimension]
6. Need to drop some attribute. The identifiers of participants are useless because the goal of the paper is to find an overall relationship between the eye gaze data and the manipulated results, which should not be constrained by specific participants properties. Also, the vote result should be removed because I want to compared the correctness of the synchronous result of human and the neuron networks.
7. Normalized the data : z-score for model1 in order to put all features into same scale or features with big scales (fixations > duration) will dominate the learning direction of the weights which will slow the training and influence the accuracy.

2.2.1 **BDNNs** [6][7][8]

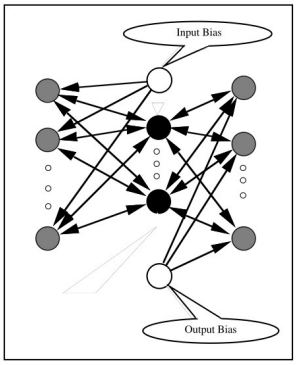


Figure 1. BDNN Topology (hidden biases not shown)

Just as the T.D. showed in the paper BDNNClassProt shows how to implement the Bi-directional network. Assume the input layer called A, Hidden layer called B and the output layer called C.

The weight A to B in the forward direction is same to the transpose of the weight B to A in the reverse direction. Similarly noticed that the forward and the reverse direction can use same weight but the output bias is only used in the forward and the input Bias is only used in the reverse. So compared to a standard MLP, the Bi-directional network has a special value input bias. Except that, the reverse direction uses the target as the output and use the linear result as the expected input to calculate the loss function.

I used one hidden layer.

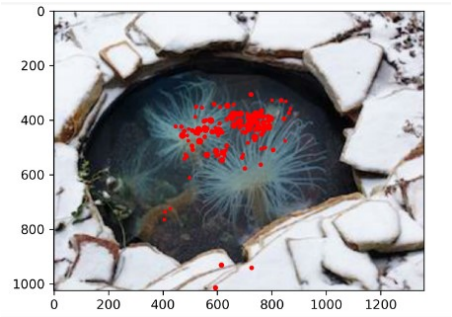
In the forward direction:	In the reverse forward:
H=relu(W1x+W1.bias)	H=relu(W2.t y+ W1.bias)
Output=sigmoid(W2H+W2.bias)	x=sigmoid(W1.t H+ W0.bias)

The hyperparameters after tuning are:

BDNNs: parameters:
W0: all 0s
n_input=9
n_hidden=10
n_output=2
batch_size=2
test_batch_size=1
epochs=50
lr=0.03
K-fold:5 fold random_state=0
Lossfunction: nll_loss
Running on cpu BDNNs.ipyon

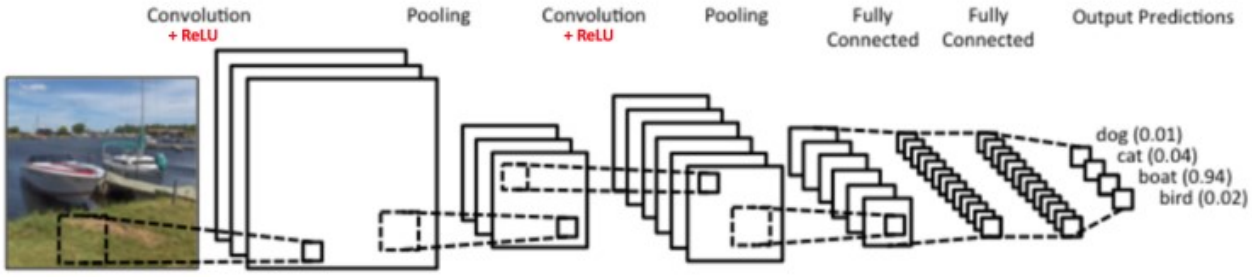
2.2.2 CNN

Model2 assume that by using the spatial data of the fixation's durations of the image. Model2 can learn the pattern of the spatial distribution of manipulated images and classify them. Chose CNN because CNN is successful in spatial problem domains in particular and my input is the 2-dimension spatial tensor: [1360 x 1024 durations]. The model2 combined CNN layers and FC layers. It used CNN layers to extract feature maps from the input. Using spatial invariant kernels to extract features from last layers. Higher level feature maps mean generic features and lower means detailed features. The feature maps will be converted into 1 dimension before entering the Fc layers. In the FC layers, using the extracted features to train a classification model.



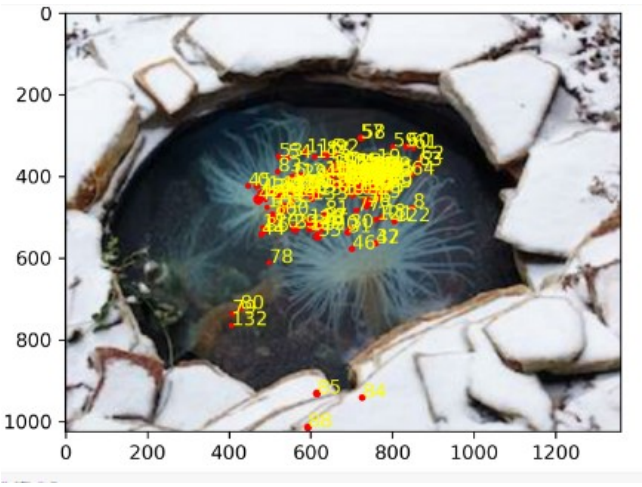
This is the format of the input. Red points have a duration>0 and the other points on this image are padding to 0. Because of the image size is 1360 x 1024, then the input is 1360 x 1024 durations for a participants observed the image 0. The input is grouped from the TimeSeries0 by participants, the output is the manipulation from Combined_image0.

[9]



The structure of the CNN is 3 convolution layers with the setting:
16 features, kernel=8, stride=4, padding=2, followed by BatchNorm2d and activation layer is LeakyRelU(0.2)
32 features, kernel=4, stride=2, padding=1, followed by BatchNorm2d and activation layer is LeakyRelU(0.2)
64 features, kernel=4, stride=2, padding=1, followed by BatchNorm2d and activation layer is LeakyRelU(0.2)
Then add 2 full connective layers：Linear1（348160>20）Linear2(20>2)
Between linear1 and linear2 is activation function relu and dropout(p=0.3) means will random drop 30% weights in order not to overfitting. The output activation function is log_softmax
The parameters after tuning:
Batch size=10 ,test batch size=1, epochs=50, lr= 0.001, momentum=0.5, weight_decay=0.0001
Weight_decay is the L2 normalize factor in order to punish the loss if the model is too complex.
Optim:SGD, loss: nll_loss
trained on gpu, colab CNN.ipyon

2.2.3 LSTM



This is the sample of sequence data of model3.
The data of model 3 is in 3 dimensions:
[Batch_size, sequence (how many words in a sentence), features dimension] in this case, features is [x,y,duration]
Because of the length of each sequence is different. It is essential to padding them in the batches. The data is not only spatial but also temporal. Model3 assumes that by knowing the order of fixations can know the manipulation. When people looking a manipulated image, it will have some particular order/logic of the fixations (e.g. people will first/ eventually keep looking at some regions on the image). LSTM is a RNN model with the forget gate to control which part need to forget and remember. It can

remember more things than normal RNN. Model 3 combined LSTM and FC layers. Model3 used the last time sequence of the LSTM as the extracted features and put it into the FC layers to classification.

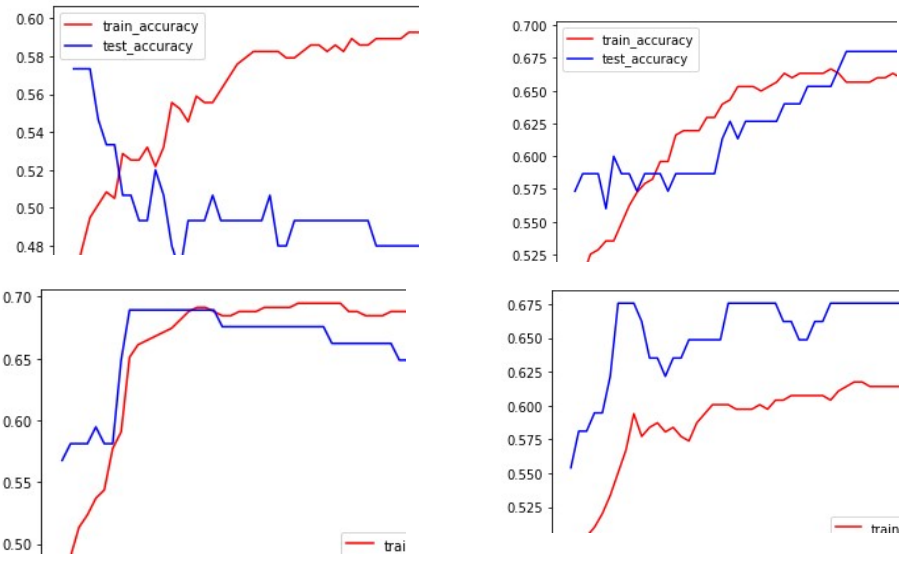
The parameters:
A two LSTM layers, 2 FC layers model.
torch.manual_seed(0) batch_first=True (need to define collate_fn of the dataloader)
input_size = 3
hidden_size = 240
linear_hidden_size=40
num_layers = 2
num_classes = 2
momentum=0
batch_size=3
weight_decay=0
num_epochs = 100
learning_rate = 0.01
dropout=0
use xavier to init the weights
f-kold:random+state=322
trained on gpu, colab
loss= .cross_entropy
optim=SGD LSTM.ipyon

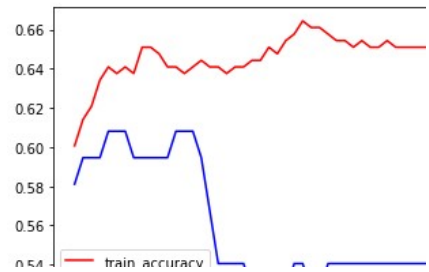
2.3 Testing:

Use the cross validation (k=5, 80% for training, 20% for testing) to test the result and draw the training and testing accuracy. Cross validation is helpful to reduce the imbalanced data distribution among training and testing datasets causing the contingency.

3 Result & Discussion

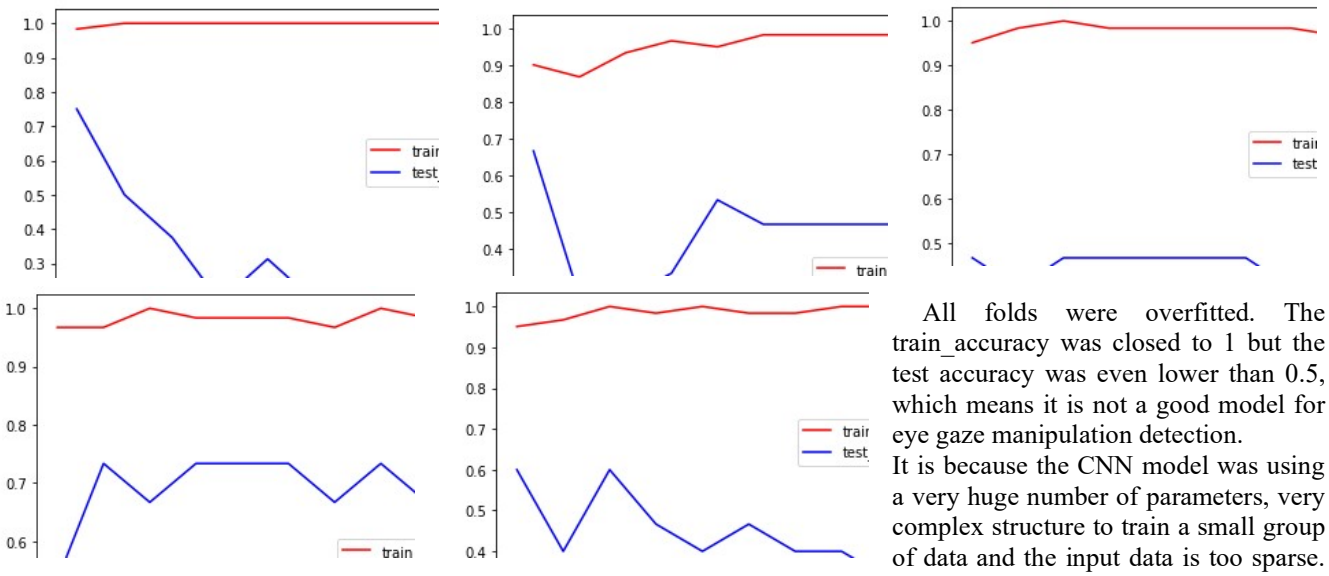
BDNNS (x=epochs, 5 folds)





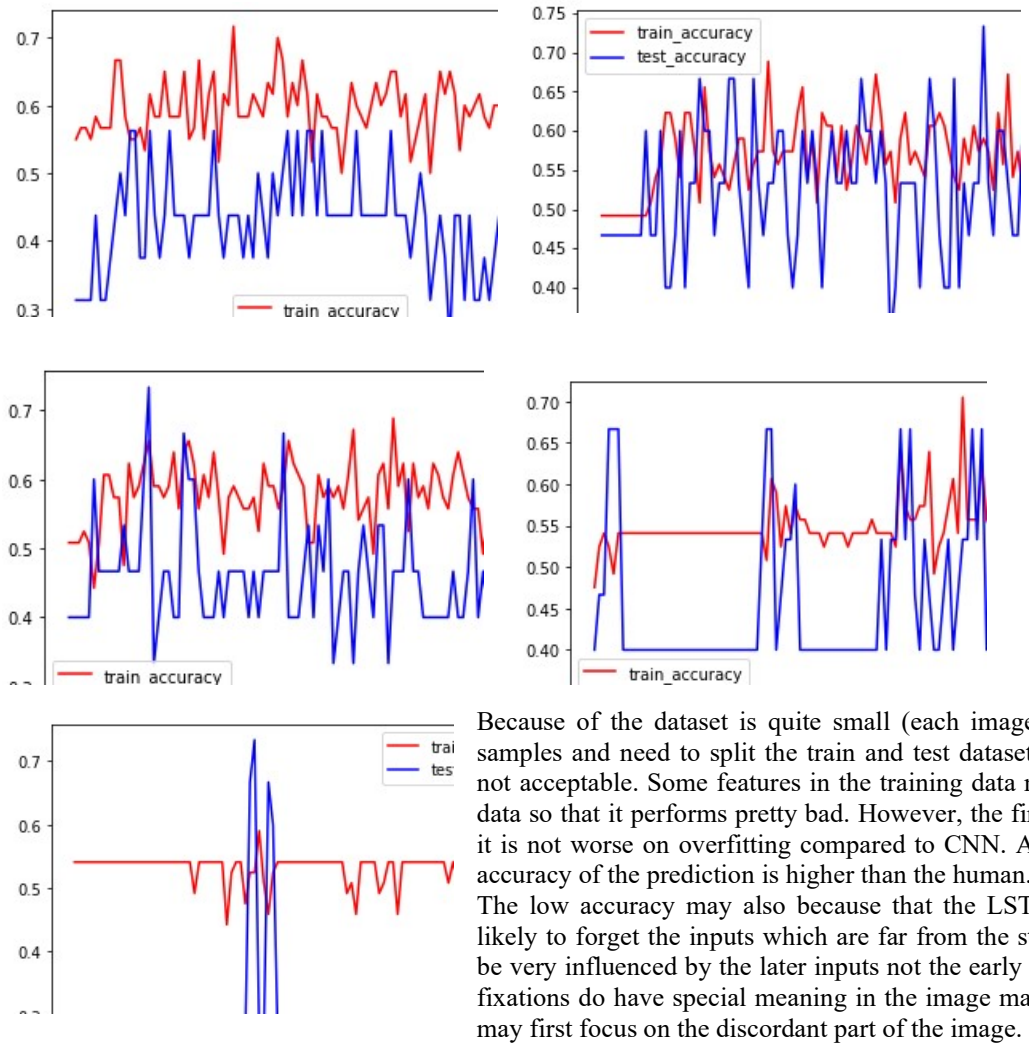
The accuracy of different k-fold was quite different in the BDNNs. The voting accuracy in Sabrina's research is 56%, which is lower than the most cases (2^{nd} ,3^{rd} ,4^{th} is higher than 60% but less than 70%). The performance of the BDNNs on the dataset of all images is generally higher than the human. However, in the 1^{st} fold, with the training accuracy increasing, the test accuracy decrease. It was overfitted in this fold because the training set was not containing enough similar samples with testing set. In the last fold, the training should be stopped early to avoid over training.

CNNs



All folds were overfitted. The train_accuracy was closed to 1 but the test accuracy was even lower than 0.5, which means it is not a good model for eye gaze manipulation detection.
It is because the CNN model was using a very huge number of parameters, very complex structure to train a small group of data and the input data is too sparse. The input form of the cnn is a sparse_to_dense tensor. The size of each image is 1360 x 1024 but actually it only contained about 100 non-zero points each image. In Sparsity Invariant CNNs [10], they said that traditional convolutional networks perform poorly when applied to sparse data even when the location of missing data is provided to the network. In order to reduce the parameters of the CNNs, we need to build sparse convolution layers to accept sparse image tensor instead of dense image tensor.

LSTM:



Because of the dataset is quite small (each image only has less than 80 samples and need to split the train and test datasets from it), the fitting is not acceptable. Some features in the training data not appear in the testing data so that it performs pretty bad. However, the first three folds show that it is not worse on overfitting compared to CNN. And in some epochs, the accuracy of the prediction is higher than the human.
The low accuracy may also because that the LSTM network would also likely to forget the inputs which are far from the start hidden state. It may be very influenced by the later inputs not the early one. However, the early fixations do have special meaning in the image manipulation area. (people may first focus on the discordant part of the image.

4 Futures:

In this research, most problems of the models came from the size of the dataset. The models can not get enough diversity samples to train and the distribution between testing and training set was not balanced. There are some ways to increase the accuracy of dataset such as using the GAN model to generates more data for training or to reduce the complexity of the model (using sparse matrix instead of dense matrix.).
For the LSTM model, by using bi-LSTM can improve the influence of the early inputs so that the model will not ignore the informative parts of the fixations.
The eye gaze data can also combine with other types data to predict the image manipulation such as linguistic data.

References

3. T.D. Gedeon Stochastic bidirectional training
[1] Hany Farid, n.d. Photo Tampering Throughout History [online] <https://www.cc.gatech.edu/~beki/cs4001/history.pdf>
[2] Grace Shao, 2019, what-is-deepfake-and-how-it-might-be-dangerous [online] <https://www.cnbc.com/2019/10/14/what-is-deepfake-and-how-it-might-be-dangerous.html >
[3] Florian Röhrbein, Peter Goddard, Michael Schneider, Georgina James, Kun Guo, (2015) How does image noise affect actual and predicted human gaze allocation in assessing image quality?
[4] Conklin, K., Pellicer-Sánchez, A., & Carrol, G. (2018). Introduction to Eye-Tracking. In Eye-Tracking: A Guide for Applied Linguistics Research (pp. 1-13). Cambridge: Cambridge University Press. doi:10.1017/9781108233279.002
[5] Sabrina Caldwell, Tamás Gedeon, Richard Jones, Leana Copeland, (2015) Imperfect Understandings: A Grounded Theory And Eye Gaze Investigation Of Human Perceptions Of Manipulated And Unmanipulated Digital Images, Barcelona
[6] A.F. Nejad and T.D. Gedeon, n.d. BiDirectional Neural Networks and Class Prototypes
[7] T. D. Gedeon, J.A. Catalan and J. Jin, n.d. Image Compression using Shared Weights and Bidirectional Networks
[8] T.D. Gedeon, n.d. Stochastic bidirectional training
[9]CS231n Convolutional Neural Networks for Visual Recognition, Stanford [online]
https://www.cnblogs.com/pinard/p/6483207.html
[10] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, Andreas Geiger (2017) Sparsity Invariant CNNs