Report of CSE 674
Project2: Explainable AI

Ankit Kumar Sinha
UBIT name:ankitsin
Person: 50286874
ankitsin@buffalo.edu

Abstract

In this project, I developed a machine learning system that learns explainable features for the handwritten words by different writers. The dataset used in this project consists of handwritten “AND”.
Our goal is to work with three different features (human determined features, deep learning features, explainable deep learning features) and compare the performance of each method for identifying whether a sample is written is by the same writer.

1 Tasks:

Following are the task performed in this project:

1. Dataset Annotation
2. Bayesian Inference of Handcrafted features
3. Deep Learning using "AND" images
4. Explainable AI- Deep Learning + PGM

2 Task 1: Dataset Annotation

For getting the 15 features of "AND" images, a online tool with feature example was provided to annotate the images. Each feature in the tool had different evaluation classes.

3 Datasets:

Following were the 3 different datset used for all the 3 different methods:

- **Seen Writer Partitioning:** In this method, train over 80% of each writers samples and test over the remaining 20% samples of each writer. Training and Validation contain images from same writer. Example if writer X had 15 images, then 12 would be in training 3 in validation.
- **Shuffled Writer Partitioning:** In this method, entire dataset is first shuffled. Training and Validation folders contain 85% and 15% of images split respectively.
- **Unseen Writer Partitioning:** In this method there exists no writer which is present in both the training and validation writer set simultaneously. Training contains 85% and 15% of writers split respectively

4 Features:

Following are the 3 different features used and extracted in this project:

1. **Human determined features**
   These have feature variables described by humans for an input image. Each feature in this has a set of discrete random variable as input.
2. **Features learned using Deep Learning**
   These consist of latent features processed by the deep learning to learn a representation using supervised or unsupervised learning.
3. **Explainable features learned using Deep Learning**
   Here also the representation are learned by deep learning is a similar manner as above and then used to create features which are similar to human determined features.

5 Task 2: Bayesian Inference of Handcrafted

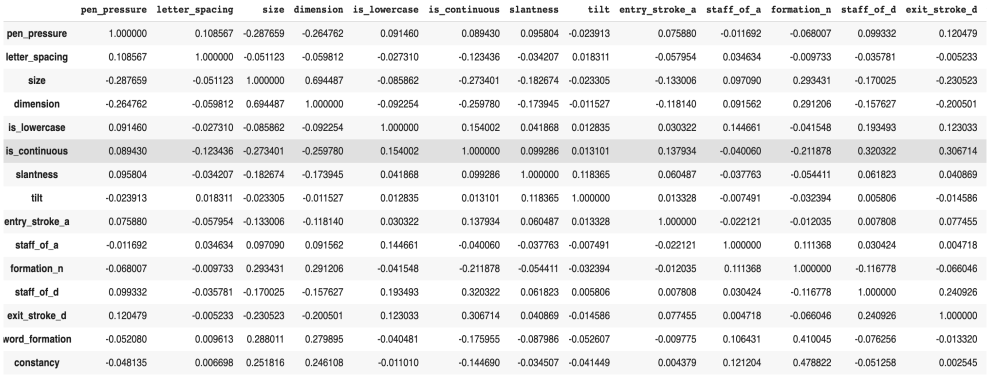
In this task, I built multiple Bayesian networks for task verification. Then compared the models on the basis of the K2 score, the number of edges and correlation matrix of features to find the best model. The best model was used to inference on the pair of images for predicting the similarity of both images and the accuracy was calculated.

5.1 Step 1: Extracting the pairs and the features

Firstly, the features and pairs for all the 3 different datasets were read. Then, using the pairs the features were paired and stored in a new dataset for input in the Bayesian network.

5.2 Step 2: Finding the correlation between features

For creating the Bayesian network correlation of the 15 features were calculated for extracting the most influencing features. Then the optimal threshold of 0.1 was searched using hit and trial method for connecting maximum nodes with min edges.



	pen_pressure	letter_spacing	size	dimension	is_lowercase	is_continuous	slantness	tilt	entry_stroke_a	staff_of_a	formation_n	staff_of_d	exit_stroke_d
pen_pressure	1.000000	0.108567	-0.287659	-0.264762	0.091460	0.089430	0.095804	-0.023913	0.075880	-0.011692	-0.068007	0.099332	0.120479
letter_spacing	0.108567	1.000000	-0.051123	-0.059812	-0.027310	-0.123436	-0.034207	0.018311	-0.057954	0.034634	-0.009733	-0.035781	-0.005233
size	-0.287659	-0.051123	1.000000	0.694487	-0.085862	-0.273401	-0.182674	-0.023305	-0.133006	0.097090	0.293431	-0.170025	-0.230523
dimension	-0.264762	-0.059812	0.694487	1.000000	-0.092254	-0.259780	-0.173945	-0.011527	-0.118140	0.091562	0.291206	-0.157627	-0.200501
is_lowercase	0.091460	-0.027310	-0.085862	-0.092254	1.000000	0.154002	0.041868	0.012835	0.030322	0.144661	-0.041548	0.193493	0.123033
is_continuous	0.089430	-0.123436	-0.273401	-0.259780	0.154002	1.000000	0.099286	0.013101	0.137934	-0.040060	-0.211878	0.320322	0.306714
slantness	0.095804	-0.034207	-0.182674	-0.173945	0.041868	0.099286	1.000000	0.118365	0.060487	-0.037763	-0.054411	0.061823	0.040869
tilt	-0.023913	0.018311	-0.023305	-0.011527	0.012835	0.013101	0.118365	1.000000	0.013328	-0.007491	-0.032394	0.005806	-0.014586
entry_stroke_a	0.075880	-0.057954	-0.133006	-0.118140	0.030322	0.137934	0.060487	0.013328	1.000000	-0.022121	-0.012035	0.007808	0.077455
staff_of_a	-0.011692	0.034634	0.097090	0.091562	0.144661	-0.040060	-0.037763	-0.007491	-0.022121	1.000000	0.111368	0.030424	0.004718
formation_n	-0.068007	-0.009733	0.293431	0.291206	-0.041548	-0.211878	-0.054411	-0.032394	-0.012035	0.111368	1.000000	-0.116778	-0.066046
staff_of_d	0.099332	-0.035781	-0.170025	-0.157627	0.193493	0.320322	0.061823	0.005806	0.007808	0.030424	-0.116778	1.000000	0.240926
exit_stroke_d	0.120479	-0.005233	-0.230523	-0.200501	0.123033	0.306714	0.040869	-0.014586	0.077455	0.004718	-0.066046	0.240926	1.000000
word_formation	-0.052080	0.009613	0.288011	0.279895	-0.040481	-0.175955	-0.087986	-0.052607	-0.009775	0.106431	0.410045	-0.076256	-0.013320
constancy	-0.048135	0.006698	0.251816	0.246108	-0.011010	-0.144690	-0.034507	-0.041449	0.004379	0.121204	0.478822	-0.051258	0.002545

Fig 1. Correlation between 15 Features

5.3 Step 3: Constructing Bayesian Network and finding the best network

Multiple Bayesian networks were created to find the optimal network. The Bayesian networks were compared on the basis of K2 score, correlation, number of edges and number of nodes.

5.4 Step 4: Fitting the model on the dataset

The model was fitted on the training dataset of all 3 types- Seen, Unseen and Shuffled. The k2 score was calculated:

Model	Seen	UnSeen	Shuffled
Model1	-2256601.812071786	-2256601.812071786	-2256601.812071786
Model2	-2063545.727606429	-2063545.727606429	-2063545.727606429
Model3	-2111951.648144342	-2069727.6393613645	-2111951.648144342
Model4	-2069727.6393613645	-2069727.6393613645	-2069727.6393613645

The best model from result of correlation, K2 Score(as K2 score is not very accurate for optimal model, it was given the least preference), least number of edges i.e 40 edges and maximum node i.e 31 nodes is **Model1**. It was used to predict on the validation dataset of all 3 types.

5.5 Step 5: Inference, Prediction and Accuracy

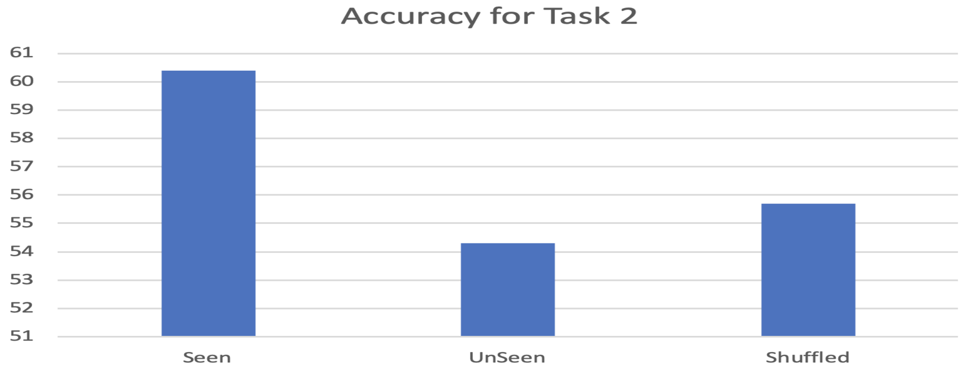
After training the model was used to predict on the validation dataset of all 3 type- Seen, Unseen and Shuffled.
For prediction, the model was used to infer the 2 queries:

- All the left image and label as 0,1 both were given as evidence to predict the right image features. Then the log likelihood was calculated for given label=0/label=1.
- All the right image and label as 0,1 both were given as evidence to predict the left image features. Then the log likelihood was calculated for given label=0/label=1.

Let a be left image, b be right image and v be label:

$$LLR = \frac{log(\frac{P(a|v=0,b=1)}{P(a|v=1,b=1)}) + log(\frac{P(b|v=0,a=0)}{P(b|v=1,a=0)})}{2} \tag{1}$$

After inferring for both sides the average was calculated to predict the label. Finally, the accuracy was calculated for the correct predicted labels.



Graph 1. Validation Accuracy for all 3 dataset for tsk2

6 Task 3: Deep Learning Inference

This task required Deep learning implementation of Auto encoder or Siamese network to extract latent features of each image and compare their similarity. As Siamese was not performing well because it can't differentiate and classify between the same writer images if it is translated. So **Autoencoder** was used for the final analysis.
It is an unsupervised learning neural network which is used to learn the latent feature representation of the input and generate back the input from the latent features.
It has two parts Encoder and Decoder.

- The encoder has multilayer neural networks like CNN and max-pooling layers which are used to reduce the dimension of the input and generate latent feature representation of it.
- The decoder has multilayer neural network like Deconvolution and Upsampling layers which are used to generate back the image from latent represenation by Deconvolving it and scaling it back to normal size.

The Autoencoder calculates the loss between input and output image and back-propagates the loss for improving the loss. It is very useful as it doesn't require target values for training the model being unsupervised and also generate the important feature which can be used to find similarity between writers.

6.1 Step 1: Create the Structure

Firstly the structure of the Autoencoder was created using the encoder and decoder. The Encoder consists of 6 convolution layer and 6 max Pooling layer which generates the 512 latent feature representation of the image which had 64*64 dimension and the decoder consisted of just the opposite layer of the encoder consisting of the deconvolution layer and upsampling to generate the image from latent features.

6.2 Step 2: Data generation and Training

As the Autoencoder can't train on all the images as input, the input is passed as batches in Autoencoder using datagen function which sends back input in batches using yield(not return) in datagen and model.fitgenerator in python. This helps in reducing the load and also stops overfitting of the model.The model was trained for 10000 epochs, 64 batch size. After training the weights were saved for future use in the project.
The model was then used to predict the latent feature representation of the input from the encoder on the validation dataset.

6.3 Step 3: Similarity between writer

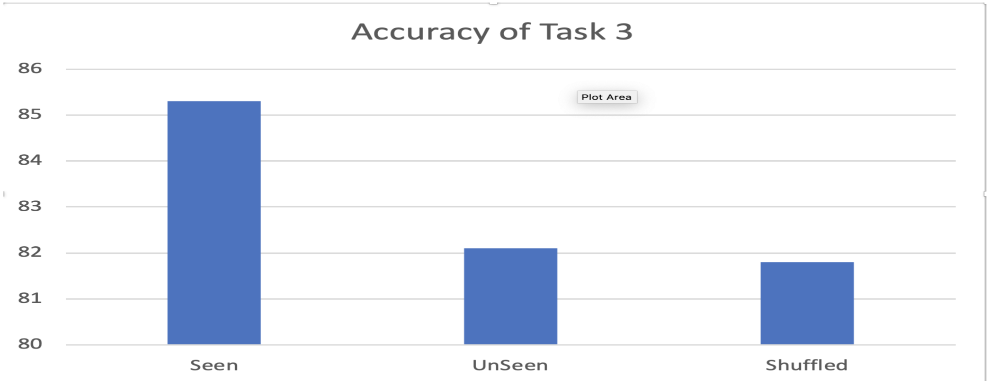
The similarity between two images is the similarity between its features and as the image original is huge a smaller representation is compared to get the similarity. Here the similarity is calculated using cosine similarity. Cosine similarity is the similarity between two vectors by calculating the cosine of the angle between them.

$$CosineSimilarity = \frac{A.B}{||A||.||B||} \tag{2}$$

As now the latent feature of the validation images are extracted, the similarity of two images is calculated using the latent feature for predicting the similarity score.

6.4 Step 4: Prediction and Accuracy

After finding the cosine similarity of all the validation image, the pairs were extracted from the validation dataset. A threshold of 0.6 was applied on the cosine similarity for predicted similar or not, the threshold was chosen when the value of false positive and false negative is nearly the same. So if the similarity is more than 0.6 its similar otherwise not similar and then the accuracy was calculated for the prediction.



Graph 2. Validation Accuracy for All 3 Dataset of Task 3

7 Task 4: Explainable AI

Explainable AI is a technique in Artifical Intelligence which explains the result predicted by the machine. Generally, the AI system and their creator cannot explain the reson behind the prediction done which is important in a very critical situation like health. So XAI is more transparent and user-friendly.
In this project, I have used the latent feature from task 3 and passed it into a multitask neural network to classify the features similar to that of Handcrafted features. After training and extracting the features from multitask neural network, it was passed into the bayesian network for predicting the similarity between writers.

7.1 Step 1: Create a Mutitask Neural Network

For creating a multitask neural network, the output from the frozen encoder i.e latent features were shared to the 15 different neural networks each consisting of 1 flatten layer, 2 dense layers and the classification ouput layer. For the 15 output of multitask, each output softmax layer is assigned a classification w.r.t to each Handcrafted features of task 2.

7.2 Step 2: Training

For the Training of this network, the loss of each layer is backpropagated to the encoder and accumulated. Then, it is passed to the encoder to updated and trained to extract the feature similar to Hand crafted features.

7.3 Step 3: Bayesian Network and Prediction

For this task, a new Bayesian network was created. This network had similarity nodes for each feature, as we dint had the regarding it, so the values for the CPD's were hard-coded. Then the inference was done on all the 30 features of both images to predict the output with the query as evidence of 30 features and prediction of similarity nodes.

7.4 Explanation

If maximum features have higher similarity score than then the writers are the same or else different, as from the fig below its visible max features are same, hence the writers are same.
For explanation, the two writer in the below fig is same as the features pen pressure, letter spacing, size, dimension, lowercase, continuous and tilt have high similarity score. So, due to these feature being same it can be reason can be explained for writer being same.

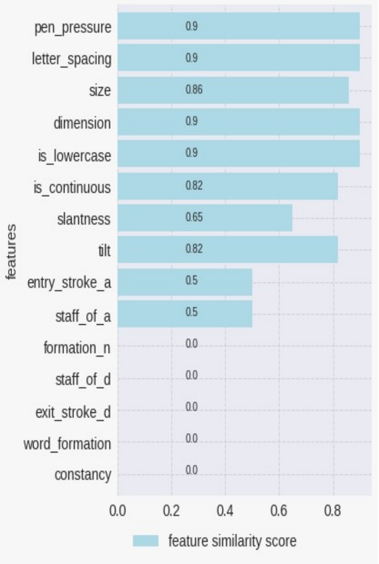
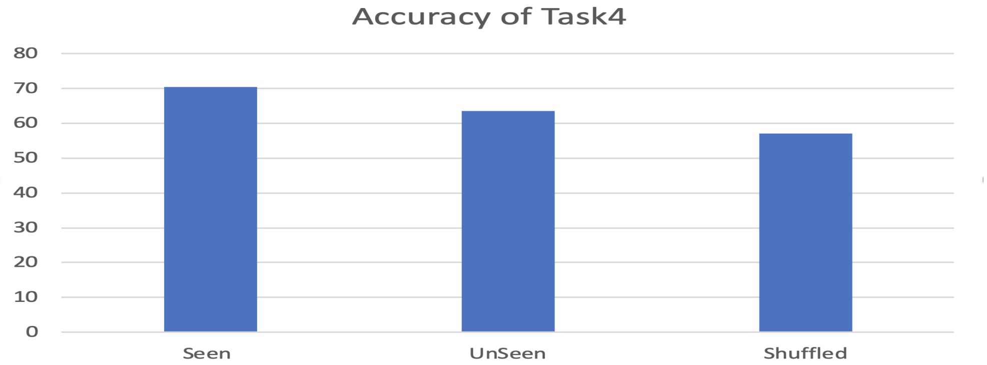


Fig 2. Similarity scores for writer.

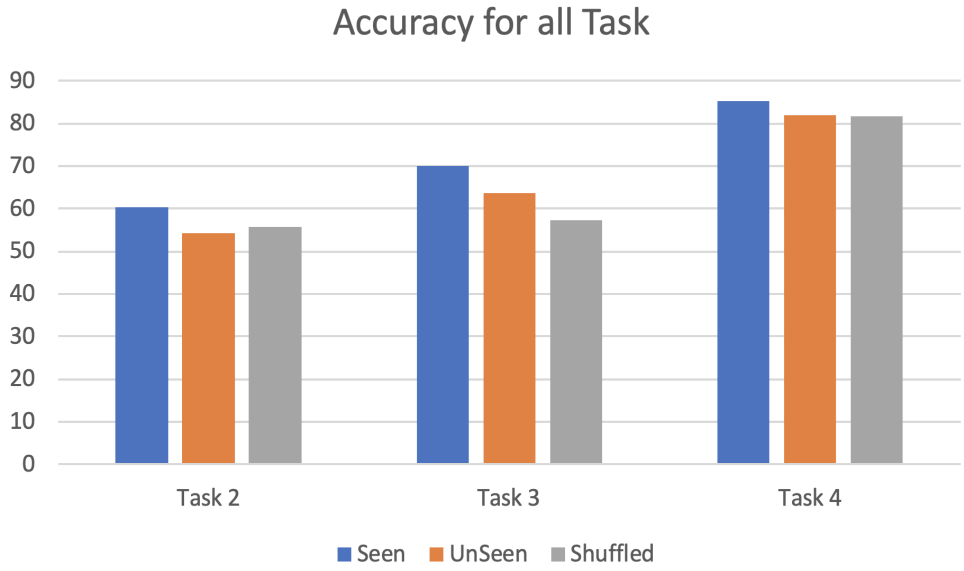
Ultimately the accuracy was calculated.



Graph 3. Validation Accuracy for all 3 dataset of Task 4

8 Final Evaluation

For all the 3 task and all 3 dataset it is clearly visible that Deep Learning performed best then XAI and lastly Bayesian network. This is very clear from the result that there is a tradeoff between accuracy and explanation, still for critical cases, explanability is required and someday the accuracy of the XAI will be same to that of Deep Learning if not better.



References

[1] https://github.com/mshaikh2/HDL$_Forensics/tree/master/AML_SPRING_2$019

[2] https://medium.com/datadriveninvestor/deep-autoencoder-using-keras-b77cd3e8be95

[3]https://towardsdatascience.com/one-shot-learning-with-siamese-networks-using-keras-17f34e75bb3d

[4]https://towardsdatascience.com/deep-inside-autoencoders-7e41f319999f *Journal of Neuroscience* **15**(7):5249-5262.