Feature Selection Using Shapley Values

Stephen V. Wright – svw2112@columbia.edu

Abstract – Feature extraction techniques reduce dimensionality of raw data by combining features into feature groups to describe the data with accuracy and originality. Gini Importance (MDI), Permutation Importance (MDA), and Actual Impurity Reduction Importance (AIR) are common measures for assessing feature importance with random forest classifier models. In this study, an alternative method of quantifying the significance of classifier features using Shapley values is tested against common classification techniques in the context of MNIST sign-language image data using the random forest algorithm. The Shapley feature selection method employs a game theoretic approach and assumes that known random characteristics should have less influence on predictions than informative features, which can be used with any Machine Learning model and allows the importance of features to be easily interpreted through a quantitative metric that reveals how each feature contributes to prediction accuracy. Unlike other single-feature selection methods, the Shapley method applies to arbitrary, user-defined groups of features and offers a useful alternative to other single-feature selection methods. In this application of Shapley values, images are split into quadrants to measure the relative importance of user-defined features and feature groups in classification problems.

Keywords – Image Classification, Feature Extraction, Game Theory, Shapley Values, Random Forest, Convolutional Neural Network, Gini Importance, Permutation Importance, Actual Impurity Reduction.

I. INTRODUCTION

With the emergence of Big Data, abstract image analysis continues to grow as a prolific field of research due to the abundance of information contained within digital images and multimedia raw data that can be extracted and applied across a variety of disciplines. Image classification forms the basis of computer vision problems and is used across all industries, including manufacturing, healthcare, agriculture, transportation, retail, and sports. For example, vision-based inspections can identify employees or travelers out of compliance with mask ordinances, or smart camera applications can be employed as a scalable method to automate visual inspections of production processes and assembly lines in industrial manufacturing. Likewise, image recognition can assist in detecting minor differences in cancerous cells, erratic behavior of livestock on smart farms, an eroding shoreline, or signs of fatigue in athletes.

Different techniques have been developed to process and analyze the large dimensionality of data extracted from images, and built-in importance values in random forest classification models are most commonly assessed using Mean Decrease Impurity (MDI) and Mean Decrease Accuracy (MDA). More recent packages such as Actual Impurity Reduction (AIR) and local interpretable model-agnostic explanations (LIME) are surrogate models trained to approximate and explain individual predictions of the underlying black box models (random forest, support vector machines, and neural networks). While each metric provides at least some degree of relative feature importance, they are limited in their ability to eliminate redundant or irrelevant features and focus the model on the most important features. Notably, the two most prominent feature importance metrics, MDA and MDG, treat features as independent and do not account for correlations or dependencies between features, which limits insight and the ability to group features to determine importance as a unit. And while optimization techniques such as principal component analysis (PCA) address the scalability issues that are common to MDA and MDG values, these methods are unable to produce insights as to why certain features are more important than others.

This study explores an alternative approach for quantifying the relative importance of image classifier features and user-defined groups of features, using Shapley values and statistical tests for a wrapper feature selection approach to determine the relevance and dependencies of features and groups of features. The Shapley feature selection method employs a game theoretic approach and assumes that known random characteristics should have less influence on predictions than informative features, which can be used with Machine Learning models and allows the importance of features to be easily interpreted through a quantitative assessment that reveals how each feature contributes to classifier accuracy.

II. RELATED WORK

Multidimensional applications of game theory to improve results in deep learning models have been prevalent in artificial intelligence research over at least the last few decades. The Nash Equilibrium in which a stable state of players cannot earn more profits by unilaterally deviating from strategies, Shapley functions where players with the highest rewards value is the most significant contributor in a group, and the Minimax theorem introduced by Von Neumann are among the most common applications of game theory in machine learning models and serves as the basis of this project. While this research borrows no code or datasets from existing studies and there are no published applications of Shapley values to quantify feature importance in sign language communications, there are certainly a number of papers that have influenced this exercise, none more than *Feature Selection Based on the Shapley Value* [1] and *Quantifying the Relative Importance of Variables and Groups of Variables in Remote Sensing Classifiers Using Shapley Values and Game Theory* [3].

A. Feature Selection Based on the Shapley Value

The Contribution Selection Algorithm (CSA) for feature selection is introduced and compared to several other existing feature selection methods, determining the usefulness of its backward elimination approach for more accurate classification results on an array of datasets with features ranging from 278 to 20,000. The algorithm is based on Shapley analysis and cooperative game theory concepts of coalitional games in which a set of players are associated with a payoff and the benefits are achieved by different sub-coalitions in a game. The Shapley-based CSA is tested against existing induction algorithms with built-in feature selection methods, using seven real-world advertising datasets to train and test the model. In four of the seven trials, the CSA algorithm outperformed other existing models with feature selection, but the random forest model without added feature selection functionality performed better with consistently higher prediction accuracy under certain conditions. Overall, the CSA demonstrated the value of applying the Shapley method with backward elimination to produce feature sets that can be used to significantly enhance classifier performance.

B. Quantifying the Relative Importance of Variables in Remote Sensing Classifiers Using Shapley Values and Game Theory

Nandlall and Millard present an alternative method that quantifies the relative importance of remote sensing classifier features and user-defined groups of features by employing concepts from game theory and Shapley values in conjunction with random forest models to produce easily-interpretable, quantifiable metrics to improve classifier accuracy. This case study goes beyond the theoretical to demonstrate the practical application of game theory and Shapley values as enhancers to existing machine learning models with built-in classifiers, using polarimetric synthetic aperture radar (SAR) images and light detection and ranging (LiDAR) data with dimensionality of 156 features to quantify the influence of sensor types, seasons, and polarimetric decomposition on the accuracy of the random forest classifier.

Within a game theoretic approach, the objective of the *game* for the three aforementioned factors is to maximize the accuracy of the land-cover classifier. The *players* are SAR-derived and LiDAR-derived groups of features. Random forest classification is run separately for all SAR features, LiDAR features, and both features combined with 10,000 trees and *mtry* parameters to obtain stable MDA values. The confusion matrices from each run were then exported and the Shapley values associated with each feature were computed. An overall increase in prediction accuracy was observed in each of the three land-cover classification scenarios, and Shapley values were consistently able to quantify how much each of the sensor, season, and polarimetric decomposition factors affect classification accuracy.

III. SHAPLEY VALUE METHOD

The approach applied in this study is modeled off a solution concept in game theory that involves fairly distributing gains and costs to several actors working in coalition, in situations when the contributions of each actor are unequal but each player works in cooperation with each other to obtain the gain or payoff. In this experiment we treat the features or groups of features as players in a collaborative game in which the objective is to maximize the accuracy of the classifier.

Consider a game with p players, and let:

- T be the set of all possible coalitions of players (2^p in all);
- $|x|$ denote the number of active players in a coalition x;
- Q(T, k) be the set of all coalitions in T that do not include player k;
- $m(x,\ k)$ denote the marginal contribution of player k to a coalition x that does not include k (that is, the score achieved by including player k in coalition x, minus the score for coalition x).

Then Shapley value for player k is:

$$S(k) = \sum_{x \in Q(T,k)} \frac{(|x|)!\,(p - |x| - 1)!}{p!} m(x,k)$$

$$= \frac{1}{p} \sum_{x \in Q(T,k)} \binom{p-1}{|x|}^{-1} m(x,k)$$

From the latter formula, the Shapley value can be interpreted as the sum of the marginal contributions of the player to all coalitions that do not include them (sum of *m(x, k)*), averaged by the number of coalitions of that size (inverse of the binomial coefficient, p-1 choose |x|), and then averaged over the number of players *p*.

IV. APPLICATION TO SIGN LANGUAGE CLASSIFICATION

The practical application of the Shapley value method was demonstrated for an application involving sign language classification. A labeled dataset from MNIST (*Figure 1*) containing grayscale 28 x 28 pixel images was used to train and test the classifier, with 27000 images used for training and 7000 images used for testing.

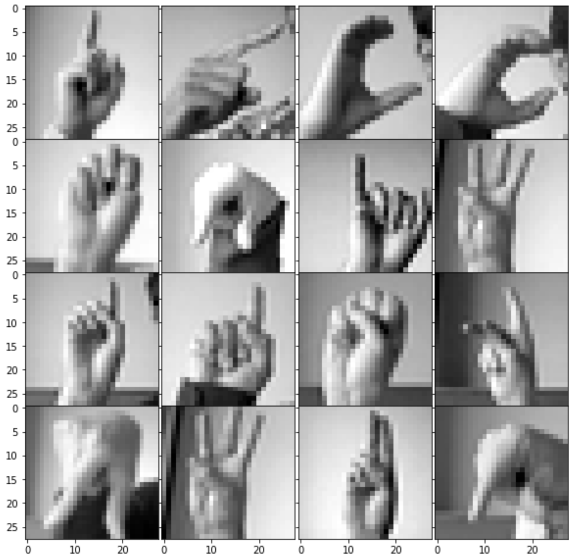


Fig 1: *Plot of Shapley values for the top-ranked individual pixel features in the random forest classifier applied to the MNIST sign language images. It can be seen that certain pixels have a higher importance than others.*

A. Random Forest and SHAP Python Packages

A random forest model was trained and tested on the dataset, returning 80.6% accuracy overall. As a baseline test for the proposed feature selection algorithm, the SHAP Python package was used to assess the relative importance of each of the 784 pixels in the images.

A plot showing the Shapley values of the top-ranked pixels is shown in *Figure 2*. The results show that some pixels in each image contribute more to classifier accuracy than others.

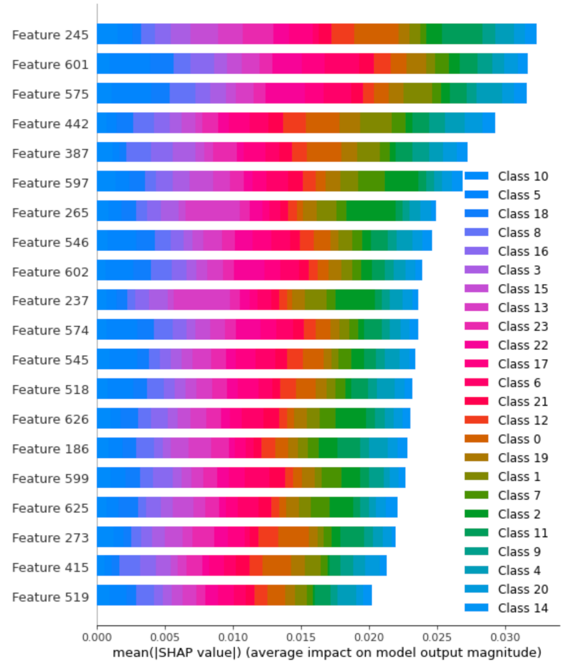
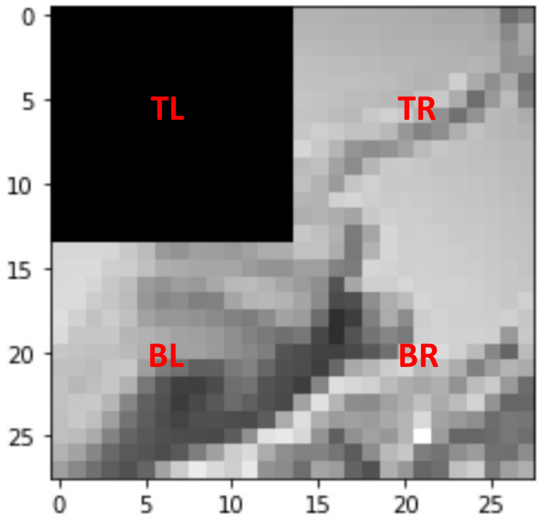
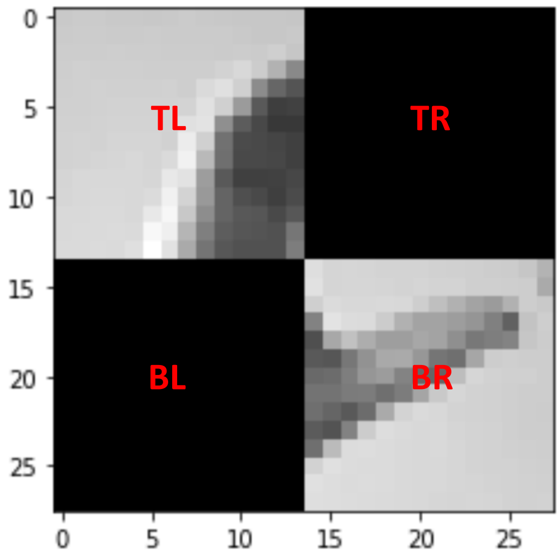


Fig 2: *Plot of Shapley values for the top-ranked individual pixel features in the random forest classifier applied to the MNIST sign language images. It can be seen that certain pixels have a higher importance than others.*

Fig 3-4: *Images divided into four quadrants for user-defined feature groups. Each quadrant can be regarded as one feature, Shapley values assess relative contributions.*





B. Shapley values for user-defined feature groups

To demonstrate how the method can be used for user-defined feature groups, each image was divided into four quadrants (top-left, top-right, bottom-left, bottom-right), as illustrated in *Figures 3* and *4*. Each quadrant can be regarded as one feature, and thus Shapley value analysis can be applied to assess the relative contribution of each to sign language recognition. The objective was to perform an initial investigation of whether certain parts of the hand contribute more than others in communicating meaning using sign language.

Random forest classifiers were trained over all combinations of groups, excluding the combination where all pixels are absent (this was assigned a baseline score of zero). This yielded a total of $2^4 - 1 = 15$ classifications. Features were removed by zeroing out the pixels in the excluded regions, as shown in Figures X.1 and X.2. In each case, accuracy was computed on the test set and recorded for each of the 15 combinations. Using these results, the Shapley values for the four features were calculated following the method in Section X. Finally, the Shapley values were averaged over three passes of the model.

C. Results

Computing the Shapley values over the full image set yields the results in Table 1. The results indicate that the top-right quadrant may have 6-7% more importance than the others (note, however, that no assessment of statistical significance has been performed here).

***Table 1**: Shapley values for each of the four quadrants, averaged over three iterations. Values sum to 80.6%, which matches classifier accuracy when all features are used.*

Feature	Shapley value
Top-left	18.0%
Top-right	25.0%
Bottom-left	18.2%
Bottom-right	19.4%

V. DISCUSSION

The classification example effectively provided an easily interpretable, unique quantitative metric when tested against the four quadrants of the split images over three iterations, suggesting that certain quadrants might be more important than others when contributing to overall prediction accuracy. While no assessment of statistical significance was tested or determined as a result of this preliminary research, the results are encouraging enough to suggest a next step in future work should include a study of statistical significance.

Likewise, future work against the MNIST sign language data would benefit from using higher resolution images or color images that allow for more specific isolation of certain pixels. Specifically, rather than applying a general classification approach by quadrant, it could be beneficial to explore how Shapley values perform in a classification example that focuses on each individual finger, in order to determine how each finger contributes to overall accuracy.

It should also be acknowledged, however, that one significant limitation of Shapley values is that the number of classifications scale exponentially. Where 10 features is manageable at $2^p - 1$ classifications, even modestly increasing feature counts would require considerable computational resources and time. Grouping features and computing Shapley values as groups is one approach to dealing with this limitation, however, if there is a desire for future research that aims to increase classifications.

VI. CONCLUSION

In this study, an alternative game theoretic approach for quantifying feature importance using Shapley values was successfully implemented to demonstrate the relative contributions of certain features in classification models. Unlike other single-feature selection methods, Shapley feature selection applies to arbitrary, user-defined groups of features and can be used with any machine learning model to replace or supplement common metrics, allowing importance features to be more easily interpreted through a quantitative assessment that reveals how each feature contributes to overall classifier accuracy.

REFERENCES

[1] Fryer, D; Strumke, I (2021). Shapley Values for Feature Selection: the Good, the Bad, and the Axioms [Online]. https:// arxiv. org/pdf /2102. 10936.pdf

[2] Rodriguez-Perez, R; Bajorath, J (2020). Interpretation of Machine Learning Models Using Shapley Values [Online]. https:// link. springer. com/content/pdf/10.1007/s10822-020-00314-0. pdf?pdf=button

[3] Nandlall, S (2019). Quantifying the Relative Importance of Variables in Remote Sensing Classifiers Using Shapley Values and Game Theory [Online]. https:// ieeexplore. ieee. org/ document/ 8718372

[4] Xiaomao, X; Xudong, Z (2019). Feature Selection Methodology for Solving Classification Problems in Finance [Online]. https:// iopscience. iop.org/article/10.1088/1742-6596/ 1284/ 1/ 0120 26/pdf

[5] Lundberg, S; Lee, S (2017). A Unified Approach to Interpreting Model Predictions [Online]. https: //arxiv.org/pdf/1705.07874.pdf

[6] Cohen, S; Ruppin, E (2005). Feature Selection Based on Shapley Value [Online]. https:// www.researchgate.net/publication/220815408_Feature _Selection_Based_on_the_Shapley_Value

COMS 4995 Deep Learning for Computer Vision

Feature Selection Using Shapley Values

Stephen V. Wright
svw2112@columbia.edu

December 09, 2022

Image Classification

- Image classification problems often involve using multiple types of digital image and multimedia data
- Volume and complexity of image data has exploded with the emergence of Big Data
- Information contained within image and multimedia raw data can be extracted and applied across disciplines
- Data types are called ‘variables’ or ‘features’; feature extraction reduces dimensionality and contextualizes data
- Use the optimal number of variables: too few = underfitting (low accuracy), too many = overfitting (more noise)

Feature Selection Methods and Limitations

- Common metrics for ranking variables include Mean Decrease in Accuracy (MDA), Mean Decrease in Gini (MGA)
- Other methods include Principal Component Analysis (PCA), Support Vector Machines (SVM), Genetic Algos
- Existing methods cannot define arbitrary groups of variables or determine the importance of each group as a unit
- Do not indicate why certain variables are important or address cases where usefulness depends on other variables
- Provide values in scaled units that can be difficult for users to interpret without optimization techniques

Shapley Values Method

- Provides a framework for measuring variable importance with an easily interpreted quantitative metric
- Relies on a metric known as the Shapley value, proposed by Lloyd Shapley in 1951 to determine fair wages
- Employs a game theoretic approach, random features have less influence on predictions than informative features
- A cooperative game, features are the ‘players’ and the objective of the game is to maximize classifier accuracy

Shapley Value Technical Definition and Properties

The Shapley value for a player is defined as:

$$\frac{1}{number\ of\ players} \sum_{coalitions\ excluding\ the\ player} \frac{marginal\ contribution\ of\ the\ player\ to\ this\ coalition}{the\ number\ of\ coalitions\ of\ this\ size\ that\ exclude\ the\ player}$$

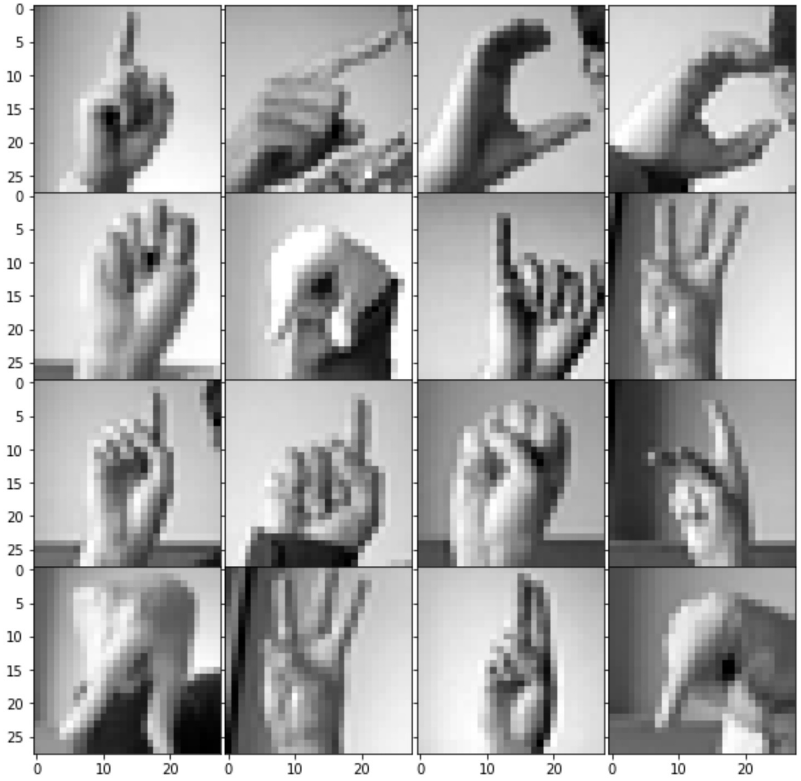
Formal definition:

- Consider a game with p players and let T be the set of all possible coalitions of players (2^p in all)
- Let |x| denote the number of active players in a coalition x
- Let $Q(T,k)$ be the set of all coalitions that do not include player k
- Let $m(x,k)$ be the marginal contribution of player k to a coalition x that does not include k
- Then the Shapley value for player k is: $S(k) = \sum_{\vec{x}\in Q(T,k)} \frac{(|x|)!(p-|x|-1)!}{p!} m(\vec{x},k) = \frac{1}{p} \sum_{\vec{x}\in Q(T,k)} \binom{p-1}{|x|}^{-1} m(\vec{x},k)$

Advantages of the Shapley Value

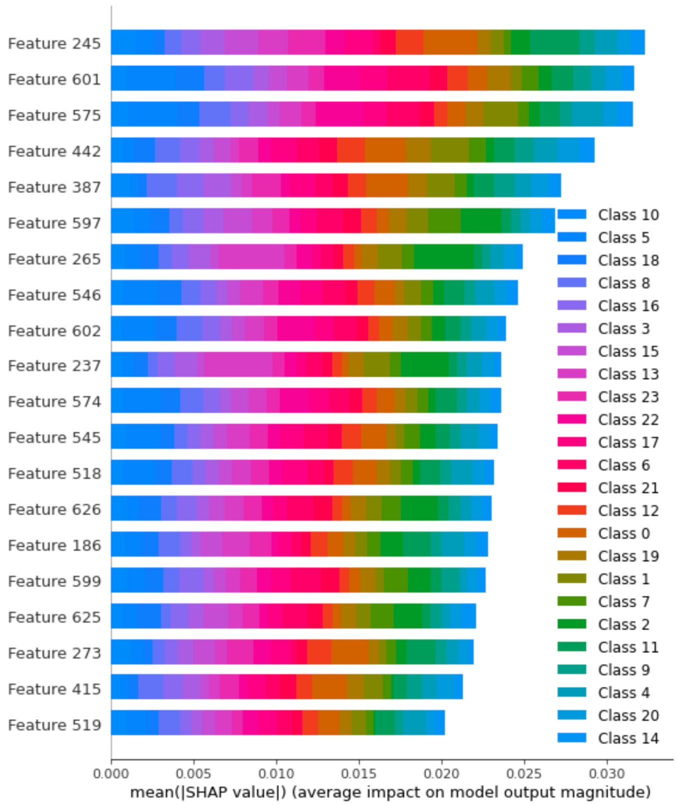
- Shapley value is defined by several properties that are desirable for a fair and just metric of player contribution
- Nondiscrimination: players with identical contributions to the game will also have identical Shapley values
- Efficiency: distributes total score across all players, the sum of which equals the score when all players participate
- Marginality: contributing players are assigned higher Shapley values, noncontributors are assigned a value of zero

Sign Language Data Classification



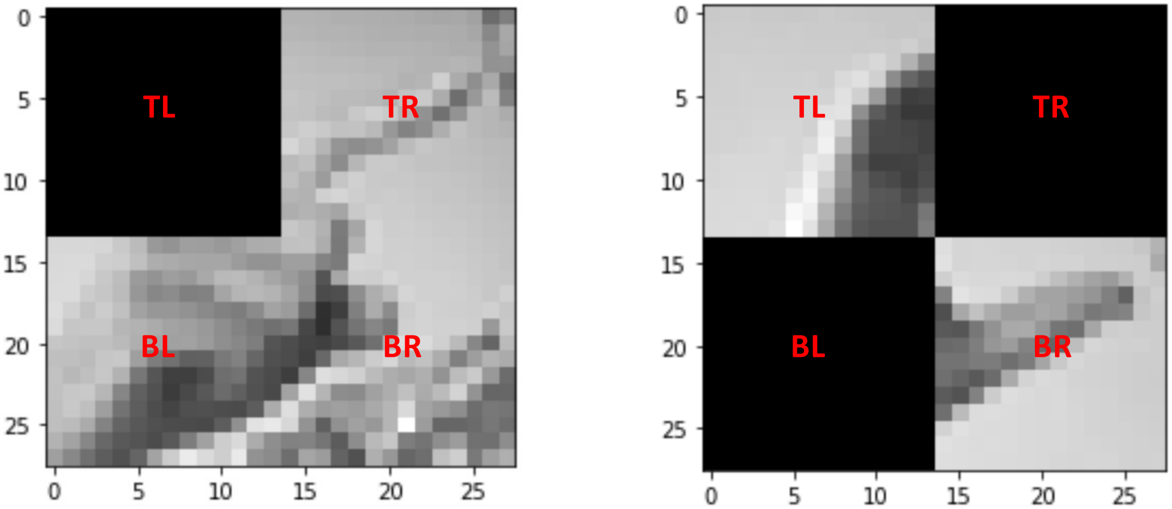
28 x 28 images

Random Forest and SHapley Additive exPlanations



Considers all 784 Pixels as Individual Features

Feature Groups



Four Quadrants of 14 x 14

Summary Results

- Random Forest python classifier with built-in feature selection returns a prediction accuracy of ~80%
- Classifier experiment covered all combinations where pixels were present, a total of $2^4 - 1 = 15$ classifications
- Shapley value results suggest that the top-right quadrant may be 6-7% more important to prediction accuracy
- Shapley Value Results: Top-left: 18.2%, Top-right: 25.2%, Bottom-left: 18.4%, Bottom-right: 19.4%

Key Findings

- Common metrics for assessing variable importance are limited in their ability to focus on important variables
- Shapley values provide a reliable framework for quantifying relative variable importance in classification problems
- Shapley values can effectively replace or supplement common metrics and the approach works with any classifier

Future Work

- Expand the dataset, further isolate sections of the photos to determine Shapley values for each finger on the hand
- Determine how much information each finger communicates in sign language, relative to other fingers in an image

References

- Fryer, D; Strumke, I (2021). Shapley Values for Feature Selection: the Good, the Bad, and the Axioms
- Rodriguez-Perez, R; Bajorath, J (2020). Interpretation of Machine Learning Models Using Shapley Values
- Nandlall, S (2019). Quantifying the Relative Importance of Variables in Remote Sensing Classifiers Using Shapley Values
- Xiaomao, X; Xudong, Z (2019). Feature Selection Methodology for Solving Classification Problems in Finance
- Lundberg, S; Lee, S (2017). A Unified Approach to Interpreting Model Predictions
- Cohen, S; Ruppin, E (2005). Feature Selection Based on Shapley Value

UNI-ID: SVW2112

Shapley value feature selection with a random forest classifier

Import libraries

In [ ]:

```
# Needed in Google Colab
!pip install shap
```

In [16]:

```
import numpy as np
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
import shap

import matplotlib.pyplot as plt
from copy import deepcopy
```

Load the dataset and show some example images.

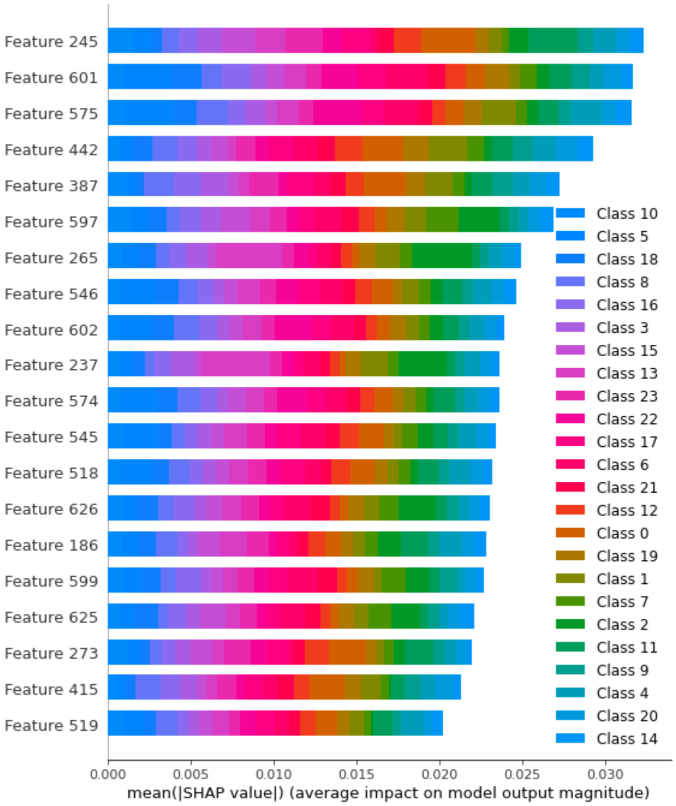
In [17]:

```
def load_data():
  train = pd.read_csv('train.csv')
  test = pd.read_csv('test.csv')

  train_labels = train['label'].values
  test_labels = test['label'].values

  train.drop('label', axis=1, inplace=True)
  test.drop('label', axis=1, inplace=True)

  num_classes = test_labels.max() + 1
  train_images = train.values / 255.0
  test_images = test.values / 255.0
  return train_images, train_labels, test_images, test_labels

train_images, train_labels, test_images, test_labels = load_data()
nrows, ncols = 4, 4
fig, axs = plt.subplots(nrows, ncols, figsize=(10, 10), sharex=True, sharey=True)
plt.subplots_adjust(wspace=0, hspace=0)

for i in range(nrows):
    for j in range(ncols):
        axs[i][j].imshow(train_images[i*ncols+j].reshape(28, 28), cmap='gray')
plt.show()
```

First-level Shapley analysis using the Python SHAP package and taking all 784 pixels as individual features. The plot obtained shows that some pixels contribute more to classifier accuracy than others.

In [10]:

```
train_images, train_labels, test_images, test_labels = load_data()
model = RandomForestClassifier()
model.fit(train_images, train_labels)
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(train_images)
shap.summary_plot(shap_values, features=train_images)
```



The results above motivate exploring the relative contribution of different regions of the image. In the example that follows, we divide the images into four quadrants and consider the relative contributions of each quadrant (top-left, top-right, bottom-left, and bottom-right). The goal will be to see if some areas of the hand (roughly speaking) contribute more to communicating than others.

In [18]:

```
train_images, train_labels, test_images, test_labels = load_data()
```

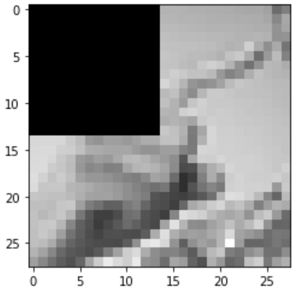
Here is an example excluding the top-left group of pixels.

In [19]:

```
F = np.zeros((28, 28))
F[0:14, 0:14] = 1
F = F.flatten()
train_images[1, F == 1] = 0
plt.imshow(train_images[1,:].reshape(28, 28), cmap='gray')
```

Out[19]:

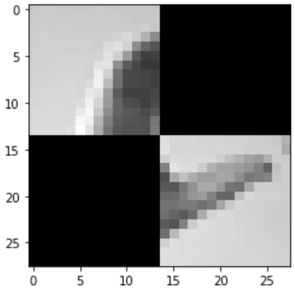
```
<matplotlib.image.AxesImage at 0x7fc0c59f58e0>
```



Here is another example excluding the top-right and bottom-left pixel groups.

```
In [20]: F = np.zeros((28, 28))
         F[0:14, 14:28] = 1
         F[14:28, 0:14] = 1
         F = F.flatten()
         train_images[2, F == 1] = 0
         plt.imshow(train_images[2,:].reshape(28, 28), cmap='gray')
```

```
Out[20]: <matplotlib.image.AxesImage at 0x7fc0c5755df0>
```



Run the classifier over all possible combinations of groups (excluding the one where all pixels are absent, which is assigned a baseline score of zero). In this case there will be a total of 2^4 - 1 = 15 classifications to run.

```
In [25]: nf = 4

         def binary_coalition(q):
           x = []
           for p in range(nf):
             r = q % 2
             q = (q - r) / 2
             x.append(r == 1)
           x.reverse()
           return x

         # Initialize with the null profile
         T = [[False for p in range(nf)]]
         acc = [0]

         for k in range(1, 2**nf):
           # Load the data
           train_images, train_labels, test_images, test_labels = load_data()

           # Generate coalition
           x = binary_coalition(k)
           T.append(x)

           # Zero out the excluded features
           F = np.zeros((28, 28))
           F[0:14, 0:14] = 1 if not x[0] else 0
           F[0:14, 14:28] = 1 if not x[1] else 0
           F[14:28, 0:14] = 1 if not x[2] else 0
           F[14:28, 14:28] = 1 if not x[3] else 0
           F = F.flatten()

           # Obscure parts of the images
           train_images[:, F == 1] = 0
           test_images[:, F == 1] = 0

           # Train and get accuracy
           model = RandomForestClassifier()
           model.fit(train_images, train_labels)
           test_labels_pred = model.predict(test_images)
           acc.append(100 * np.sum(test_labels_pred == test_labels) / len(test_labels))
```

Compute and display the Shapley values (shown in the order of top-left, top-right, bottom-left, and bottom-right).

In [26]:

```
# Compute Shapley
def get_profile_index(x, T):
  for k in range(len(T)):
    if T[k] == x:
      return k
  return -1

S = []
for p in range(nf):
  # Track marginal accuracies by coalition size
  marginal_acc = [[], [], [], []]

  # Loop over all input profiles in the universe of the game
  # that do NOT containt player p
  for x in T:
    if not x[p]:
      # Find the size of the input profile
      n = np.sum(x)

      # Find base accuracy (without the player contributing)
      acc_base = acc[get_profile_index(x, T)]

      # Find the accuracy of the same coalition with the player now contributing
      v = deepcopy(x)
      v[p] = True
      acc_add = acc[get_profile_index(v, T)]

      # Add marginal accuracy
      marginal_acc[n].append(acc_add - acc_base)

  # Count coalition sizes
  #print(marginal_acc)
  V = 0
  for k in range(nf):
    V += np.mean(marginal_acc[k])
  V = V / nf
  S.append(V)

print(S)
```

```
[17.84021193530396, 24.71649005391337, 18.149284253578735, 19.22987544153188]
```

Here we see that the top right quandrant of the sign language images are more important than other quadrants in contributing to prediction accuracy.

A more interesting example might be to determine Shapley values for each finger of the hand, which would provide an estimate of how much information each finger communicates in sign language, relative to the other fingers. The main barrier to doing this is obtaining a good quality dataset: likely, a higher resolution than 28 x 28 pixels would be desirable, and it would also require labour intensive work of identifying and tracing the individual fingers of the hand in each training image.

Access to full datasets and presentation video can be found at the following Google Drive location:

https://drive.google.com/drive/folders/1pPR8L4Wl07o8QMf3xprcozO_XcGHfuWC?usp=sharing (https://drive.google.com/drive/folders/1pPR8L4Wl07o8QMf3xprcozO_XcGHfuWC?usp=sharing)