Final Project Report

Veracity

Jain Devansh Rakesh (CH17B050)
RA Keerthan (CH17B078)

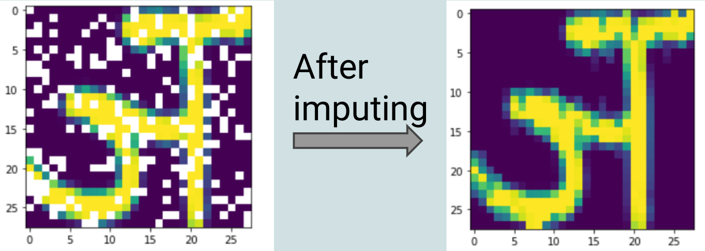
Problem Statement

- The objective is to classify handwritten characters of Devanagari script. Each character belongs to a class.
- Training set of 1000 images belonging to 10 classes were provided where every training sample has missing pixels which are to be imputed.
- Test set comprises of 1000 images belonging to 10 classes.

Solution approaches

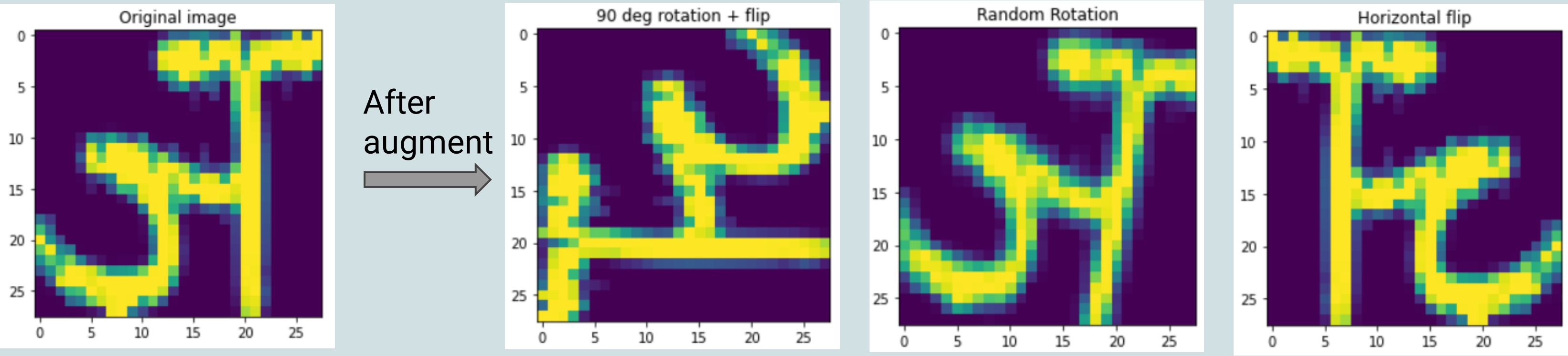
Preprocessing

- Missing values filled using interpolatation and KNNImputer(). Tuned parameter n_neighbors.



After imputing

- Image Augmentation: Rotated the image by 90 degrees and flipped it vertically. Random rotation is also imposed on the original images along with horizontal flipping.



After augment

Original image

90 deg rotation + flip

Random Rotation

Horizontal flip

Solution approaches

Motivation:

To analyze the effect of image augmentation and compare the performances of SVM, kNN and Random Forest (RF) with and without PCA.

Approach 1

- Training set: Training set without augmentation.
- Model: kNN with cross validation and gridsearch to tune hyperparameters.

Approach 2

- Training set: Training set without augmentation.
- Model: SVM with cross validation and gridsearch to tune hyperparameters.

Approach 3

- Training set: Training set without augmentation.
- Model: RF with cross validation and gridsearch to tune hyperparameters.

Solution approaches (contd..)

Approach 4

- Training set: Training set with all the augmentations.
- Model: PCA with number of components that explain atleast 75% of the variance + kNN with cross validation and gridsearch to tune hyperparameters.

Approach 5

- Training set: Training set with all the augmentations.
- Model: PCA with number of components that explain atleast 75% of the variance + SVM with cross validation and gridsearch to tune hyperparameters.

Approach 6

- Training set: Training set with all the augmentations.
- Model: PCA with number of components that explain atleast 75% of the variance + RF with cross validation and gridsearch to tune hyperparameters.

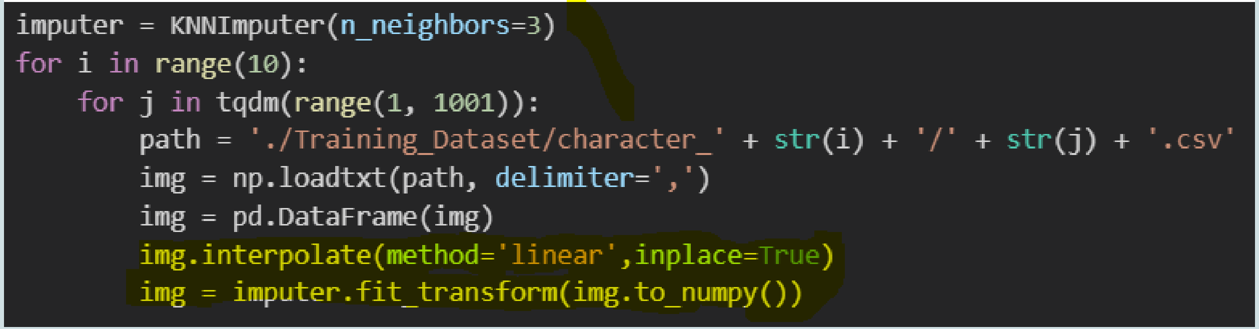
Approach 7

- Training set: Training set with all the augmentations.
- Model: PCA with number of components that explain atleast 75% of the variance + Voting classifier of SVM, kNN and RF.

Best Solution

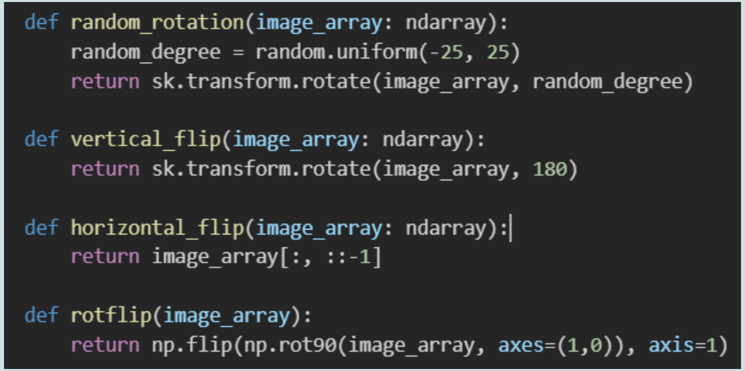
Data Preprocessing:

- Imputation of missing pixels was done with pandas ***linear*** interpolation followed by kNN Imputer (***n_neighbors*** parameter was tuned and set to ***3***).



```
imputer = KNNImputer(n_neighbors=3)
for i in range(10):
    for j in tqdm(range(1, 1001)):
        path = './Training_Dataset/character_' + str(i) + '/' + str(j) + '.csv'
        img = np.loadtxt(path, delimiter=',')
        img = pd.DataFrame(img)
        img.interpolate(method='linear',inplace=True)
        img = imputer.fit_transform(img.to_numpy())
```

- Image augmentation with random, horizontal and vertical rotations and flips or a combination of both was performed to artificially expand training data.



```
def random_rotation(image_array: ndarray):
    random_degree = random.uniform(-25, 25)
    return sk.transform.rotate(image_array, random_degree)

def vertical_flip(image_array: ndarray):
    return sk.transform.rotate(image_array, 180)

def horizontal_flip(image_array: ndarray):
    return image_array[:, ::-1]

def rotflip(image_array):
    return np.flip(np.rot90(image_array, axes=(1,0)), axis=1)
```

Best Solution (contd..)

Train Test Split, PCA and SVM:

- The original training data was shuffled and split into train and validation data in a stratified fashion in a ratio of 80:20.
- PCA (with components that explain atleast 75% variance) was fit on the train data and both train and validation data were transformed.
- SVM model with RBF kernel was trained and GridSearchCV with 3 fold cross validation was used to tune the following SVC parameters:

  ```
  {'gamma':['scale','auto'], 'shrinking':[True,False], 'class_weight':[None,'balanced'], 'C':[1,10,100]}
  ```

- We got the best parameters as:

  ```
  {'gamma':'scale’, 'shrinking':True, 'class_weight': 'balanced', 'C':10}
  ```

- Finally PCA(0.75) was fit on original train data and then transformed. SVC with the best parameters found earlier was trained on this data and saved as a pickle file.

Results

Approach	**Score on test-set**
no Augmentation (noAug) + kNN	0.216
noAug + SVM	0.233
noAug + Random Forest	0.228
PCA + Aug + kNN	0.968
PCA + Aug + SVM	**0.971**
PCA + Aug + RF	0.956
PCA + Aug + Voting Classifier	0.955

Conclusion

- Looked at how imputing is done by combining linear interpolations and kNN imputer.
- Compared the effect of image augmentation and dimensionality reduction on kNN, SVM and RF.
- Image augmentation and PCA improved performance.
- Using SVM as the classifier yielded the highest score on test set.

References

1. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.interpolate.html
2. PCA, kNN, SVM and RF from Sklearn documentation.
3. https://medium.com/@thimblot/data-augmentation-boost-your-image-dataset-with-few-lines-of-python-155c2dc1baec