DIABETIC RETINOPATHY DETECTION USING SVM

Team Members:
Koushik Rameshbabu (2020103019)
Pranav Prshanth(2020103554)
Batch: P, Semester: 6

Base Paper

Diabetic Retinopathy Detection and Classification Using Mixed Models for a Disease Grading Database.

Link: https://ieeexplore.ieee.org/document/9343812/

Data Feasibility

The dataset should be diverse enough to include images from a variety of sources, including different camera types, lighting conditions, and patient populations. This ensures that the detection system can generalize well to new and unseen data. The data Should also be large enough so that it can be filtered accordingly. Keeping in mind that high-definition camera are being used in the process of getting the images and dataset size may vary from 500Gb to 1Tb and also can be further filtered with help of MapReduce

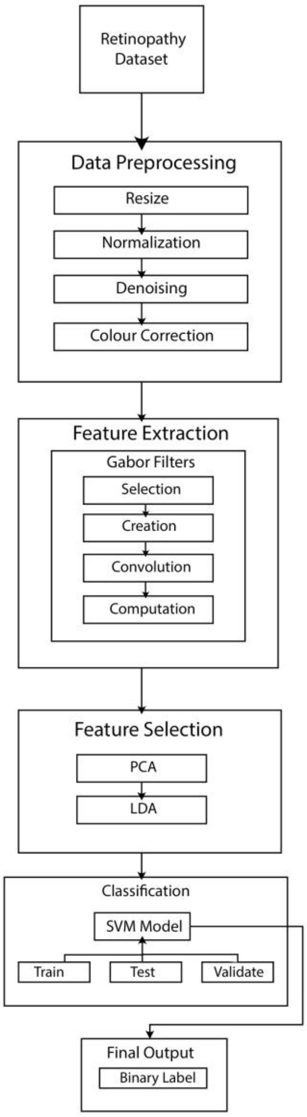
OBJECTIVE

The objective of Diabetic Retinopathy (DR) Detection and Classification using Support Vector Machines (SVM) is to develop a machine learning model that can accurately identify the presence and severity of DR in retinal images. Diabetes frequently results in diabetic retinopathy, which damages the blood vessels in the retina and can result in blindness or vision loss. Accurate identification and diagnosis of the disorder are essential since early discovery and prompt treatment can stop or slow the progression of DR. An effective machine learning algorithm for classification tasks is SVM. SVM can be trained to distinguish between normal and aberrant retinal pictures and to categorise the severity of DR into several phases, such as mild, moderate, or severe, in the case of DR detection and classification.

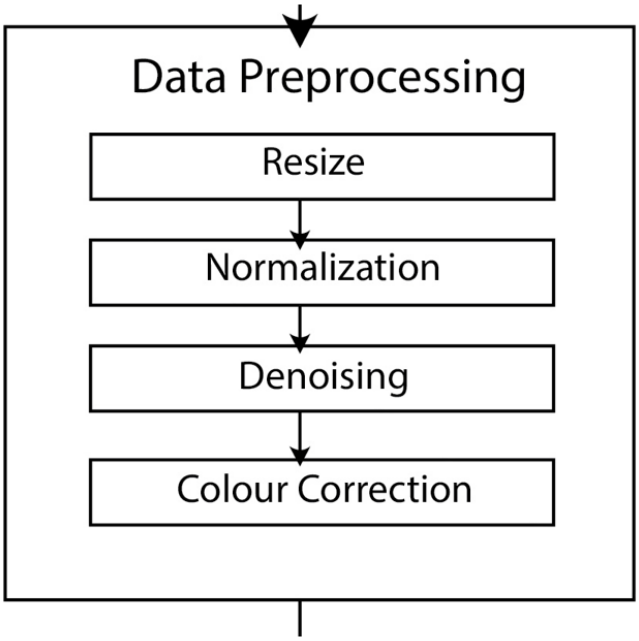
Problem Statement:

The problem statement involves in addressing the challenges of accurately identifying and classifying different stages of DR, such as mild, moderate, and severe, using SVM. It also involves optimizing the SVM model to ensure high accuracy and efficiency in diagnosing DR from retinal images. The successful development of an SVM-based DR detection and classification model can significantly improve the efficiency and accuracy of DR diagnosis, enabling early detection and timely treatment, ultimately reducing the risk of vision loss in patients with diabetes.

Overall Architecture:

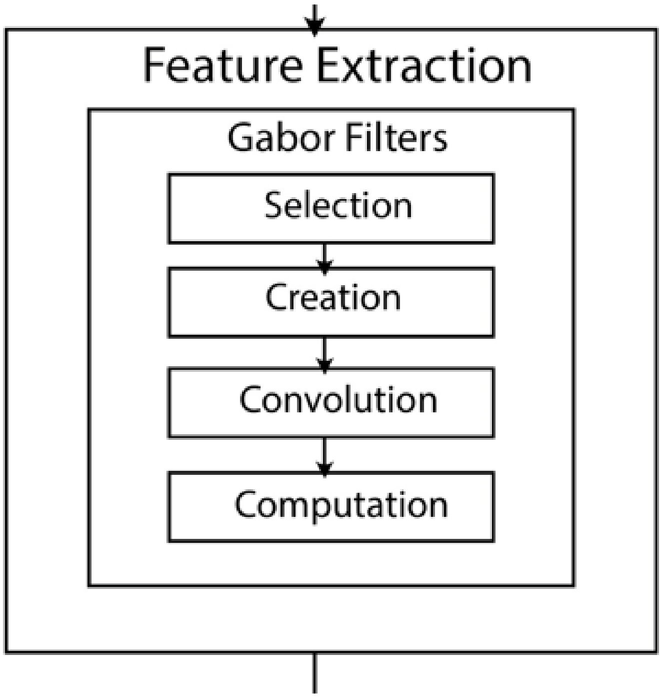


Modules Descriptions



The first step is to acquire high-quality retinal images of patients with and without DR. The images are typically obtained using specialized cameras or ophthalmoscopes. The images are then resized to a standard resolution to ensure consistency across the dataset. Normalizing the extracted features can help to reduce variations and improve accuracy. Feature selection techniques can also be applied to remove irrelevant features and reduce the complexity of the SVM model.

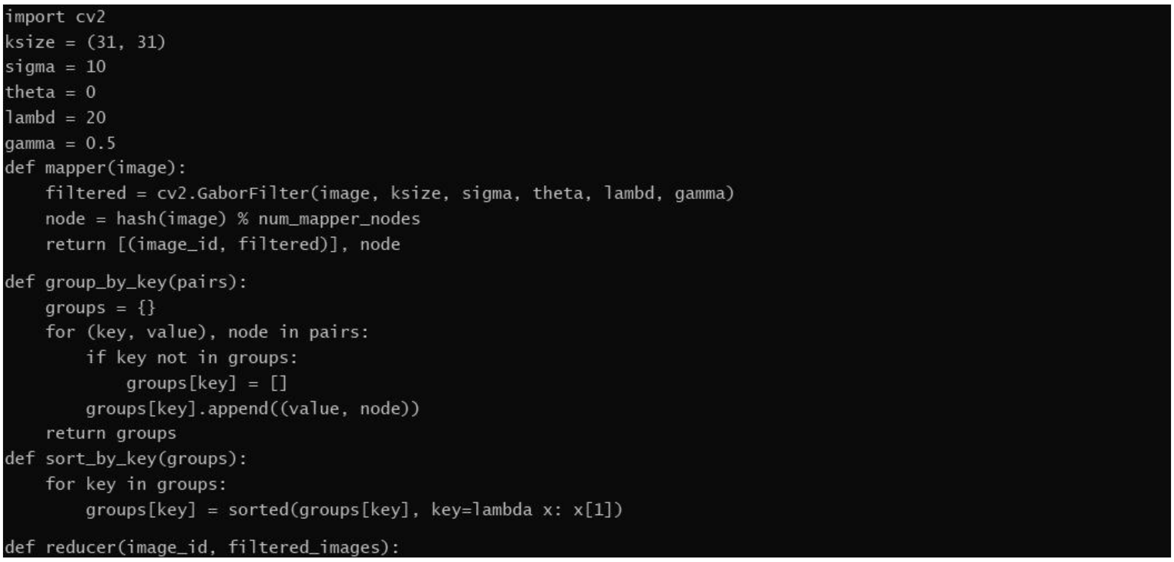
Retinal images can be affected by variations in lighting and contrast, which can affect the accuracy of DR detection. Pre-processing techniques such as contrast stretching and histogram equalization can be applied to enhance image quality and reduce variations. Overall, data pre-processing is critical in the development of a precise and efficient SVM-based DR detection and classification model. Proper pre-processing techniques can aid in improving model accuracy and lowering the risk of false positives or false negatives in DR diagnosis.



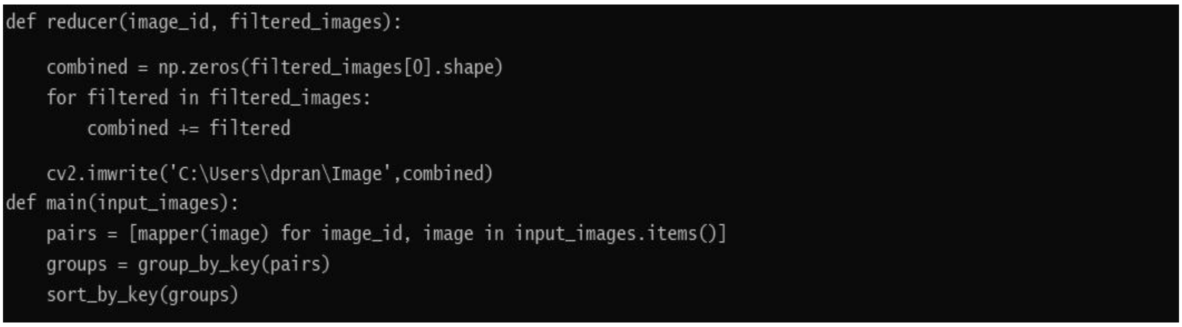
Extracting informative features from retinal images is a critical step in developing an accurate DR detection and classification model. Features can be extracted using techniques such as edge detection, texture analysis, and wavelet analysis and Gabor Filters. In this project the feature extraction method that is being used is the Gabor filter along with MapReduce.

1. Map phase:
- Read input retinal images and extract features (e.g., texture, edge, wavelet).
- Assign each image to a mapper node for processing.
- Map each image to a set of (key, value) pairs, where the key is the SVM model (e.g., mild, moderate, severe) and the value is the feature vector.
2. Shuffle and Sort phase:
- Group the (key, value) pairs by the SVM model.
- Sort the (key, value) pairs within each group by the key.
3. Reduce phase:
- For each group of (key, value) pairs, train an SVM model using the feature vectors as input and the SVM model as the target output.
- Apply the trained SVM models to classify new retinal images.

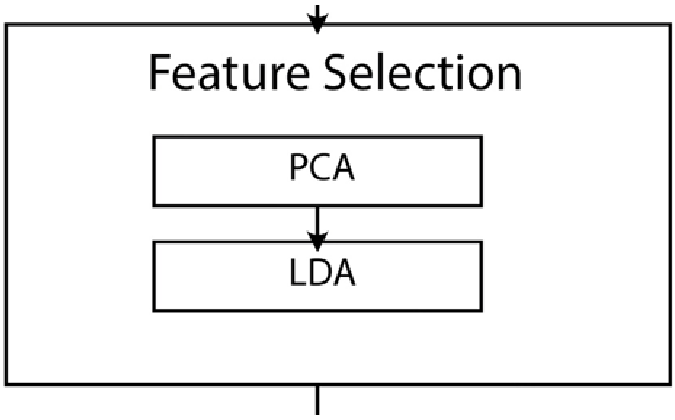
MR Code for Gabor Filter



```
import cv2
ksize = (31, 31)
sigma = 10
theta = 0
lambd = 20
gamma = 0.5
def mapper(image):
    filtered = cv2.GaborFilter(image, ksize, sigma, theta, lambd, gamma)
    node = hash(image) % num_mapper_nodes
    return [(image_id, filtered)], node

def group_by_key(pairs):
    groups = {}
    for (key, value), node in pairs:
        if key not in groups:
            groups[key] = []
        groups[key].append((value, node))
    return groups
def sort_by_key(groups):
    for key in groups:
        groups[key] = sorted(groups[key], key=lambda x: x[1])

def reducer(image_id, filtered_images):
```



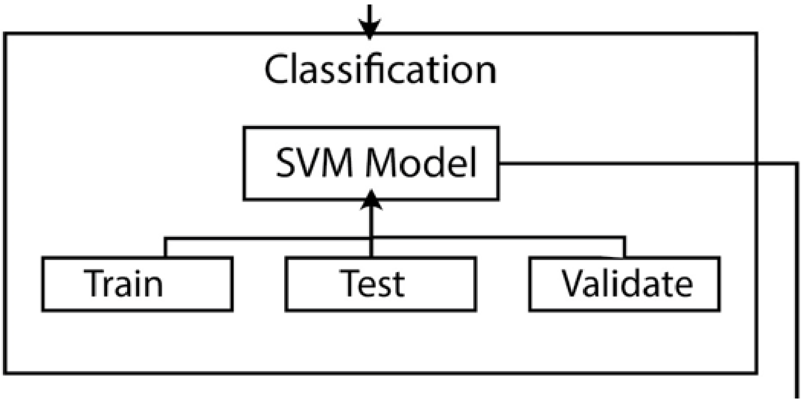
```
def reducer(image_id, filtered_images):

    combined = np.zeros(filtered_images[0].shape)
    for filtered in filtered_images:
        combined += filtered

    cv2.imwrite('C:\Users\dpran\Image',combined)
def main(input_images):
    pairs = [mapper(image) for image_id, image in input_images.items()]
    groups = group_by_key(pairs)
    sort_by_key(groups)
```



The selection of features is an important step in the detection and classification of diabetic retinopathy using SVM. The goal of feature selection is to identify a subset of features that are most relevant to the classification task while reducing feature space dimensionality and avoiding overfitting. The two-feature selection technique used here are the principal component analysis (PCA) and Linear discriminant analysis (LDA).
The main goal of PCA is to transform the original features into a new set of orthogonal features called principal components.

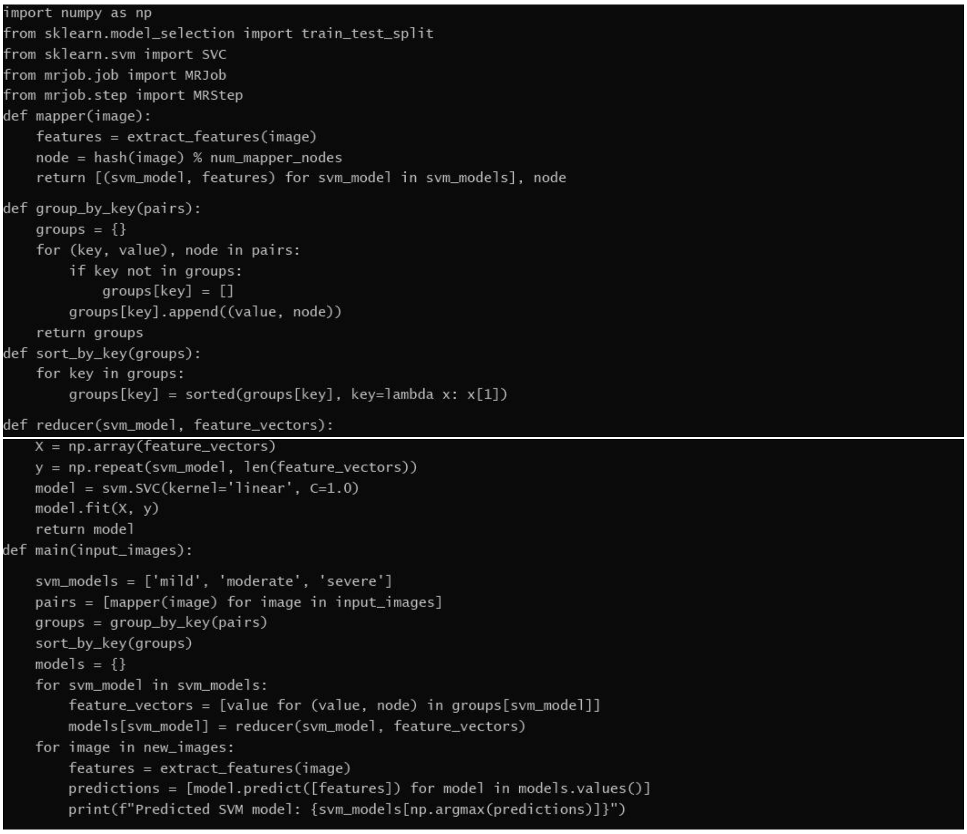
PCA can be used to reduce the dimensionality of the feature space while retaining as much information as possible. This can help to overcome the curse of dimensionality and improve the performance of the SVM model. PCA is particularly useful when there are many correlated features in the dataset, as it can identify the underlying structure in the data and reduce the redundancy among features. Linear discriminant analysis, is a commonly used technique for dimensionality reduction and feature selection in Diabetic Retinopathy Detection and Classification using SVM.

The main goal of LDA is to find a linear combination of features that best separates the classes in the data. LDA works by projecting the original feature space onto a lower-dimensional space while maximizing the separation between the classes. The resulting feature space is chosen to maximize the between-class variance and minimize the within-class variance.



It is important to note that the hyperparameters should be tuned using cross-validation on the training set to prevent overfitting. Additionally, the performance metrics on the testing set should be reported to ensure that the SVM model can accurately classify unseen data. After that Use the trained SVM model to predict the class labels of the test dataset. Evaluate the performance of the SVM model on the test dataset using metrics such as accuracy, precision, recall, F1-score, and ROC curve. This can be done with the help of MR code.

MR code for SVM



```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from mrjob.job import MRJob
from mrjob.step import MRStep
def mapper(image):
    features = extract_features(image)
    node = hash(image) % num_mapper_nodes
    return [(svm_model, features) for svm_model in svm_models], node

def group_by_key(pairs):
    groups = {}
    for (key, value), node in pairs:
        if key not in groups:
            groups[key] = []
        groups[key].append((value, node))
    return groups
def sort_by_key(groups):
    for key in groups:
        groups[key] = sorted(groups[key], key=lambda x: x[1])

def reducer(svm_model, feature_vectors):
    X = np.array(feature_vectors)
    y = np.repeat(svm_model, len(feature_vectors))
    model = svm.SVC(kernel='linear', C=1.0)
    model.fit(X, y)
    return model
def main(input_images):

    svm_models = ['mild', 'moderate', 'severe']
    pairs = [mapper(image) for image in input_images]
    groups = group_by_key(pairs)
    sort_by_key(groups)
    models = {}
    for svm_model in svm_models:
        feature_vectors = [value for (value, node) in groups[svm_model]]
        models[svm_model] = reducer(svm_model, feature_vectors)
    for image in new_images:
        features = extract_features(image)
        predictions = [model.predict([features]) for model in models.values()]
        print(f"Predicted SVM model: {svm_models[np.argmax(predictions)]}")
```

Justification For Using MR

The datasets used in Retinopathy Detection are frequently large and complex, making traditional computing methods difficult to process. The dataset can be divided into smaller chunks and processed in parallel using MapReduce, allowing for efficient processing of large datasets. Traditional computing methods can take a long time to process large datasets. By parallelizing the processing across multiple computing nodes, MapReduce can significantly reduce processing time.

GaborFilterMapReduce class in the code that extends the MRJob class is provided by the mrjob library. The mapper method reads in each block of the input image, converts it to grayscale, applies Gabor filter to it using a bank of Gabor filters, and yields the output block. The reducer method combines the output blocks into the final output image. Finally, the steps method defines the MapReduce steps to execute

Overall, the use of MapReduce with SVM can enable efficient and scalable machine learning for tasks such as Retinopathy detection, where large datasets and complex models are involved.