**Introducation**

Automatic face and gender recognition has become relevant to an increasing amount of applications, particularly since the rise of social media. Performance of existing methods on new images is often considered a significant task, especially when compared to the immense leaps in performance reported for the related task of face recognition.

Reinforcement learning (RL) is a machine learning technique for solving sequential decision problems. Traditionally, a single basis is used to represent an image for classification instead of using multiple bases, which are a major problem with feature extraction using learning, where feature vectors are extracted from a suitable and learned basis set via Principal Component Analysis (PCA), and the agent then learns to extract optimal feature basis from this feature space through RL. Therefore, the environment of this RL problem becomes feature space, and, selected feature, reward is defined to help the agent reinforce the subset of features that have high representational powers and to punish those that represent poorly.

Many previous works proposed different feature representations and an RL framework for solving this problem. However, the collected features in the original paper, features, eigenfunctions, eigenpictures namely, every thing that is on and used interchangeably for a given learning process to be efficient, each time step is updated with a novel feature vector, where current state is defined to be last feature vector used in the previous update, which is a standard approach in RL literature.

The representation (Q-values) of the environment in this work is used to classify images represented simply in different feature spaces.

**Pseudo Code**

**Algorithm OLB Learning**

Select randomly an OLB OLB(x, y, T) from the training dataset (OLB(x, y, T), OLB(x, y, T), …, OLB(x, y, T), OLB(x, y, T)).

Initialize the corresponding Q-tables randomly.

for iteration = 1 to number of Episodes do

h = 0, 0, 0, …

repeat

Select a feature index a_t = f_t by greedy policy.

Update the selected feature vector, f_t = f_t - 1.

for all the representation points X_j, j = 1, 2, …, m in training dataset into space defined by h using p_j = dist(x, h).

Find the class labels of the K-nearest neighbors of the projected data from p_j.

until the corresponding Q-value of the step, the agent receives the maximum reward or if all the features are the selected.

eend for

**Proposed Work**

**Pace 2: PCA - Feature Pools**

The main idea behind the PCA-process is to use the set of eigenvector images, extract features corresponding to the feature maps in the image, apply the following to these data. 1. End the process if the image is within the eigenvector space, otherwise project the image to the eigenvector space first before proceeding. 2. End the process if the image is within the eigenvector space, otherwise project the image to the eigenvector space first before proceeding. 3. Extract eigenvalues and dimensions of eigenvectors, i.e., (λ_1, λ_2) of the eigenpicture, with its corresponding eigenvalue. 4. Set weights of images into eigenvector space can be calculated as a representative by taking each eigenvector component, for instance, the eigenvector of the image of the eigenvector space.

We have isolated each image as an eigenvector of OLB(x, y, T), where T is the representation point of OLB, which is the weight vector, representing an image, k is the class of the image, and T is the vector of class of C is the optimal feature basis of the current stage, and it is initialized to 0.

**Pace 1: OLB-Learning**

The learning process is applied to each OLB(x, y, T), where the classification problem is modeled using the 50176-labeled images from the ORL database. We have chosen this feature representation because of its ability to capture important structural information in the images. The proposed classification is the same as the one proposed in the original paper, which means that the optimal feature basis of the current stage has the highest Q-value in the corresponding eigenvector. It's a classification, a model that is trained using multiple feature vectors, each with a different number of features. The model is trained using the corresponding eigenvectors, which are all of the same type, and the one that has the highest Q-value in the corresponding stage is chosen as the optimal feature basis of the current stage.

**Future work**

In the future, I intended to explore with different image datasets that are more consistent within different groups, such as MNIST, ORL, and RUT70.

Furthermore, due to a lack of computational power for PCA-based methods, I am interested in a more efficient implementation of PCA that can be run on standard desktops. Future work will involve implementing and evaluating different PCA-based methods on standard desktops, and exploring the feasibility of implementing PCA-based methods on mobile devices.

**Literature cited**


