

Mục lục

ĐỒ ÁN PHÂN LOẠI ẢNH FRUITS	2
I. GIỚI THIỆU TỔNG QUAN	2
1. Các dạng rút trích đặc trưng	2
2. Các loại thuật toán học máy.....	2
3. Bài toán phân lớp:	4
4. Bộ dữ liệu thực nghiệm:	4
5. Mô hình phân loại ảnh.....	5
II. THỰC NGHIỆM CÁC LOẠI RÚT TRÍCH TRONG MATLAB VÀ ĐÁNH GIÁ KẾT QUẢ	6
1. Histogram of oriented gradients (HOG)	6
2. Create Bag of Visual Words (BoW).....	8
Đánh giá kết quả phân lớp: đếm số lượng mẫu đúng và xuất kết quả.....	9
3. Machine learning methods: HoGFeatureKNN, HoGFeatureSVM.....	11
3.1. HoGFeatureKNN:.....	11
3.2. HOGFeartureSVM:.....	12
3.3. So sánh các features và thuật toán cho mô hình phân loại ảnh	18
III/Kết luận	19
IV/Tài liệu tham khảo	19

BÁO CÁO ĐỒ ÁN DẠNG 2

ĐỒ ÁN PHÂN LOẠI ẢNH FRUITS

1. Link Source Code và báo cáo

<https://github.com/trint11/Nopbaocao.git>

2. Link Video Kết quả chạy demo

<https://drive.google.com/drive/folders/1-im5970Rg7QF5mecd8VmZPRH-LEQnPl>

I. GIỚI THIỆU TỔNG QUAN

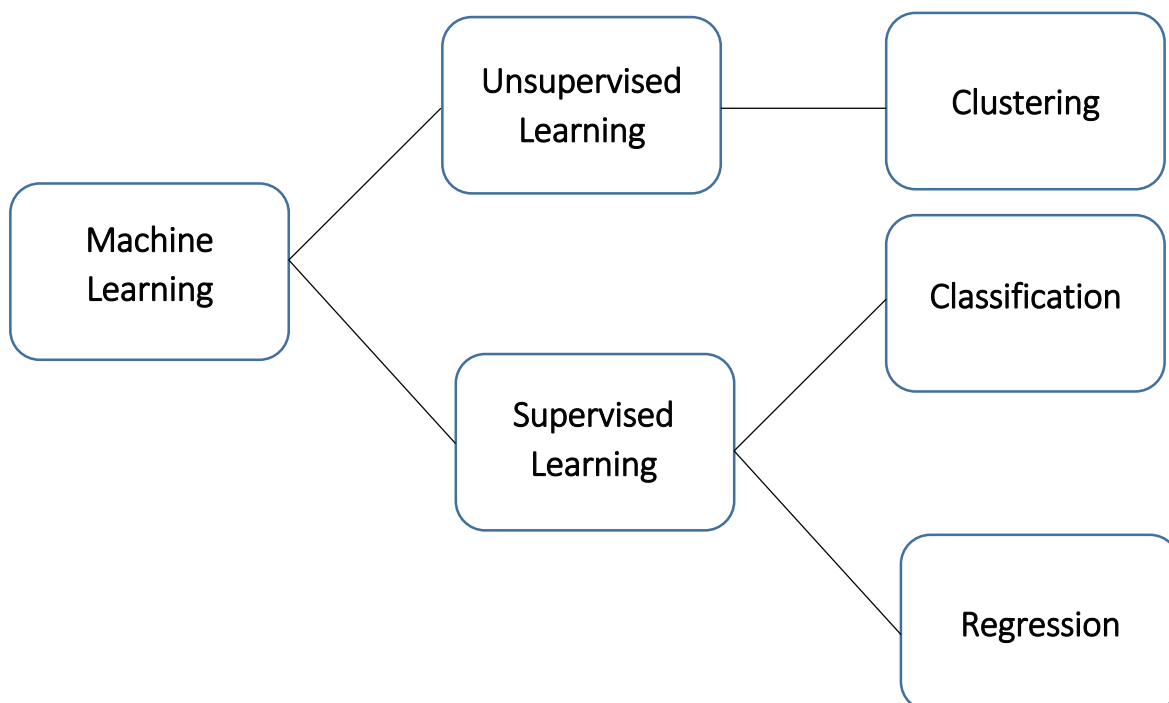
1. Các dạng rút trích đặc trưng

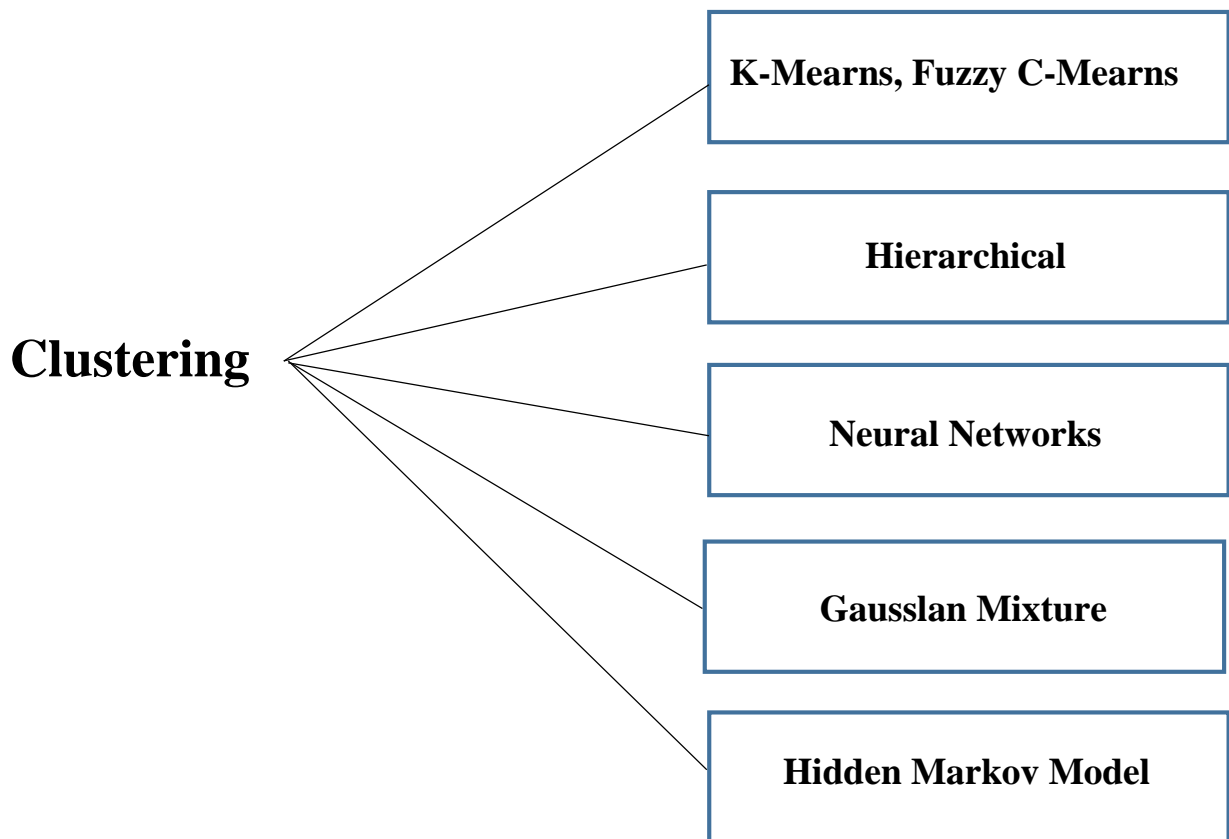
- Histogram of oriented gradients (HOG)
- Local binary patterns (LBP)
- Alexnet (Deep Learning)
- Bag of visual Words (BoW)
- Haar wavelets
- Color histograms, ect

Ngoài ra, còn có các dạng phát hiện các điểm đặc trưng như: SURF, HARRIS, ect

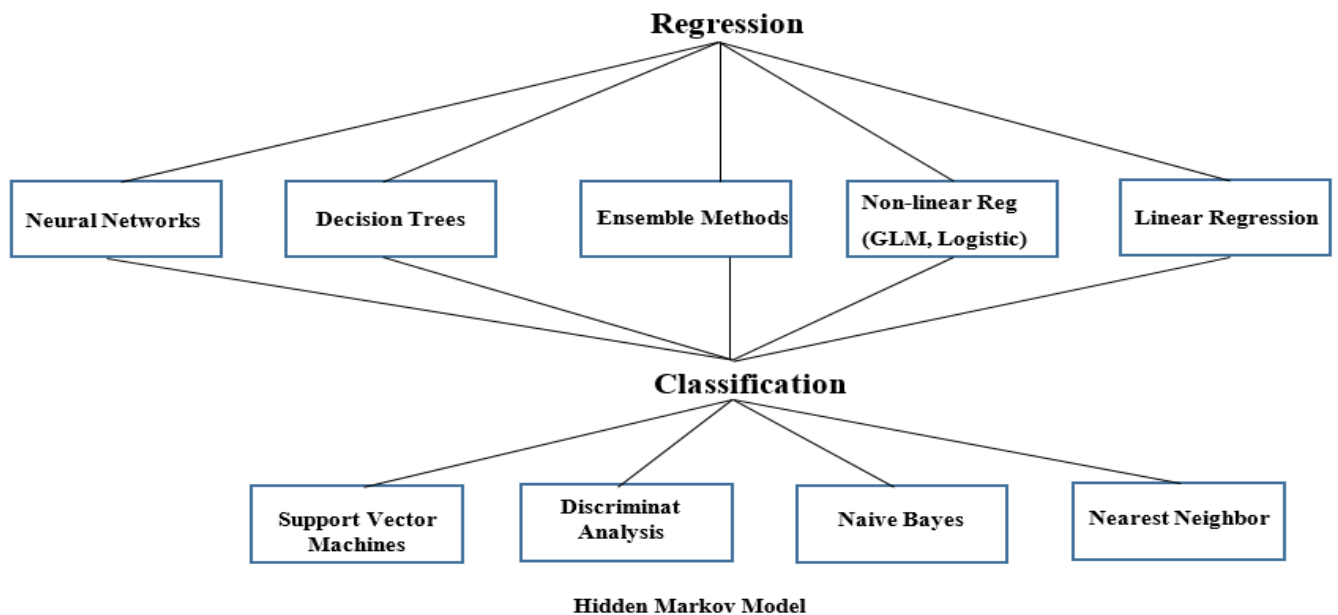
2. Các loại thuật toán học máy

Các mô hình tổng thể về các thuật toán học máy như bên dưới



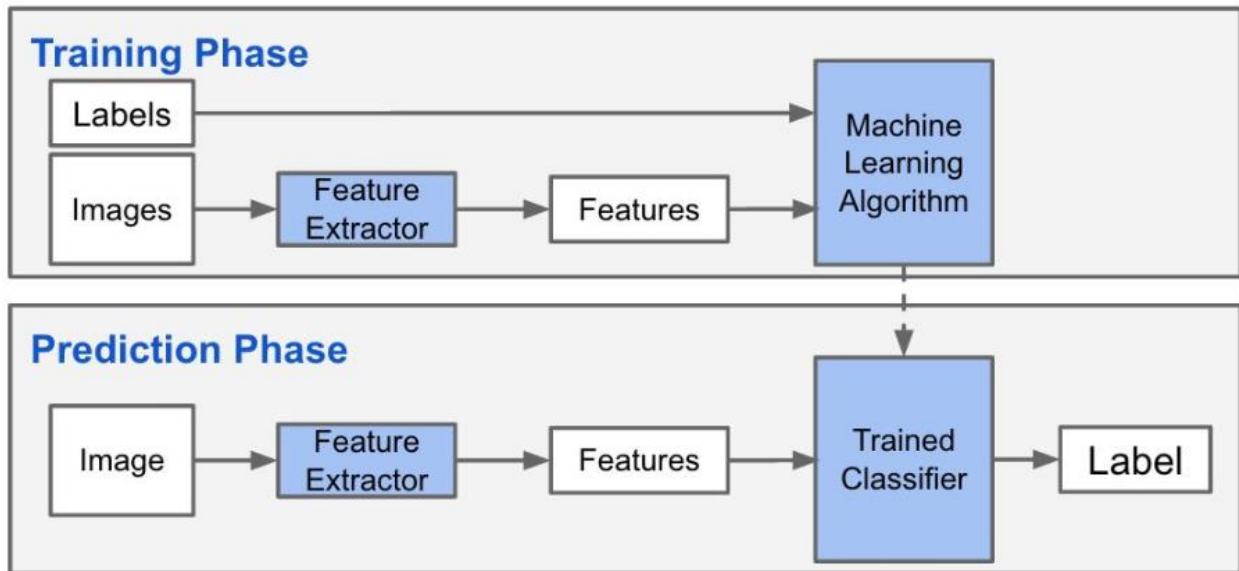


Supervised Learning



Ngoài các thuật toán như mô hình tổng thể trên còn có thuật toán khá nổi tiếng gần đây đó là Deep Learning.

Mô hình bài toán VRA



Trong các mô hình nhận dạng thị giác đa phần chúng trải qua các bước sau đây:

- Xây dựng bộ dữ liệu data train
- Huấn luyện dữ liệu (load data train)
- Rút trích đặc trưng (chọn một loại rút trích phù hợp)
- Xây dựng model (chọn một thuật toán học máy phù hợp)
- Nạp dữ liệu test (load data test, có thể là Image hoặc video)
- Dựa vào model để dự đoán kết quả và độ chính xác.

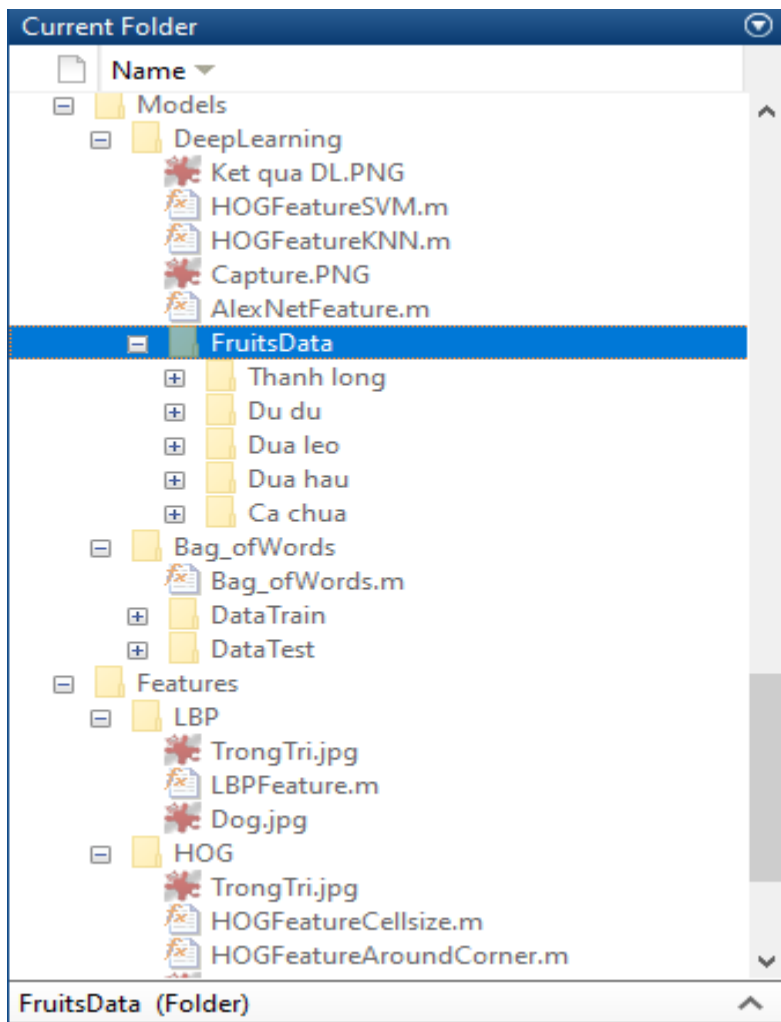
3/Bài toán phân lớp:

Để mô tả cho bài toán phân lớp chúng tôi xin trình bày phần này bằng một bài toán:

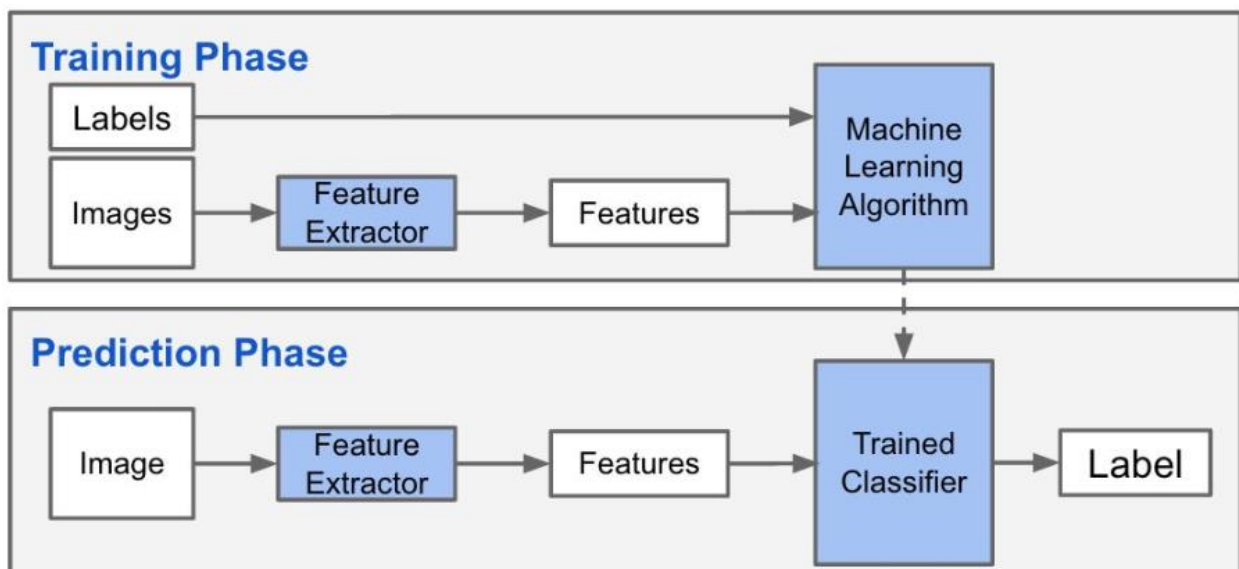
4/Bộ dữ liệu thực nghiệm:

Bộ dữ liệu thực nghiệm do chúng tôi tạo ra, chúng có một thư mục gốc FruitsData và 5 thư mục con, mỗi thư mục đại diện cho một lớp đối tượng trong đó chứa 50 image và được tổ chức như hình sau

Kích thước image: 227x227 fixel



5/ Mô hình phân loại ảnh



Trong đồ án phân loại ảnh, để đánh giá mô hình chúng tôi kết hợp rút trích đặc trưng và thuật toán học máy lần lượt có tên như sau:

- Mô hình BoW: như đã giới thiệu phần trên (BoW với KNN và SVM)
- Mô hình HOG: sử dụng rút trích đặc trưng Histogram of oriented gradients (HOG) và thuật toán K-nearest neighbor (k-NN)
- Mô hình HOG: sử dụng rút trích đặc trưng Histogram of oriented gradients (HOG) và thuật toán Support Vector Machine (SVM)
- Mô hình Alexnet: sử dụng rút trích đặc trưng Alexnet và thuật toán Support Vector Machine (SVM)
- Mô hình Alexnet: sử dụng rút trích đặc trưng Alexnet và thuật toán Support Vector Machine (k-NN)

II. THỰC NGHIỆM CÁC LOẠI RÚT TRÍCH TRONG MATLAB VÀ ĐÁNH GIÁ KẾT QUẢ

1/ Histogram of oriented gradients (HOG)

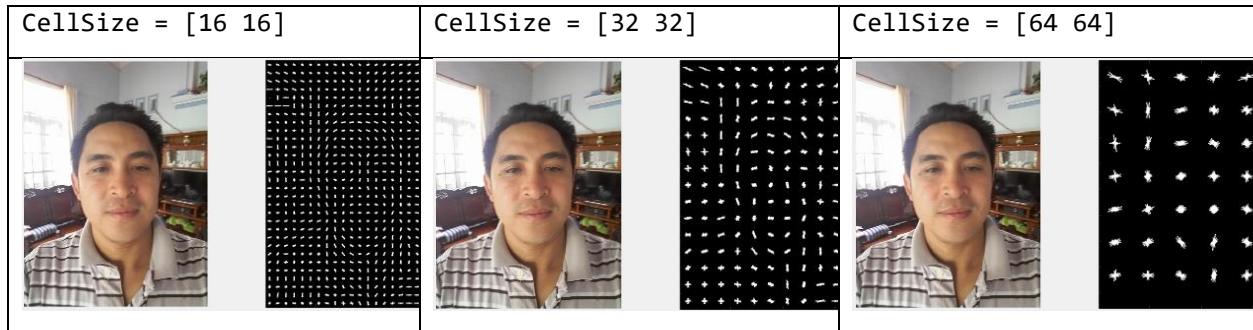
a/ Giới thiệu mô hình trong đồ án

- Nạp dữ liệu train của bài toán nhận dạng
- Lấy đặc trưng HOG cho từng ảnh trong dữ liệu train.
- Xây dựng một model từ tập dữ liệu đặc trưng HOG
- Nạp dữ liệu test của bài toán nhận dạng
- Sử dụng model để predict cho tập dữ liệu test.

b/ Thực nghiệm các loại rút trích đặc trưng trong Matlab

```
1. function HOGFeatureCellsize()
2. %Read the image of interest.
3. I1 = imread('TrongTri.jpg');
4. %Extract HOG features.
5. [hog1,visualization] = extractHOGFeatures(I1,'CellSize',[128 128]);
6. %Display the original image and the HOG features.
7. subplot(1,2,1);
8. imshow(I1);
9. subplot(1,2,2);
10.    plot(visualization);
11.    end
```

Thay đổi cellsize ta lần lượt thu được các kết quả như sau:

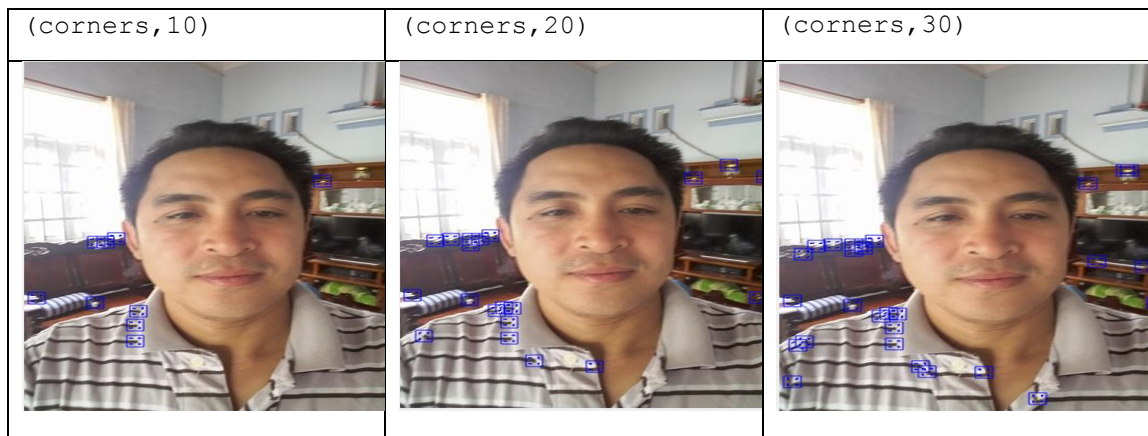


```

1. function HOGFeatureAroundCorner()
2. %Read in the image of interest.
3. I2 = imread('TrongTri.jpg');
4.
5. %Detect and select the strongest corners in the image.
6. corners = detectFASTFeatures(rgb2gray(I2));
7. strongest = selectStrongest(corners,30);
8.
9. %Extract HOG features.
10. [hog2, validPoints,ptVis] = extractHOGFeatures(I2,strongest);
11.
12. %Display the original image with an overlay of HOG features
    around the strongest corners.
13. figure;
14. imshow(I2);
15. hold on;
16. plot(ptVis, 'Color', 'blue');

```

Thay đổi corner lần lượt thu được các kết quả như sau:



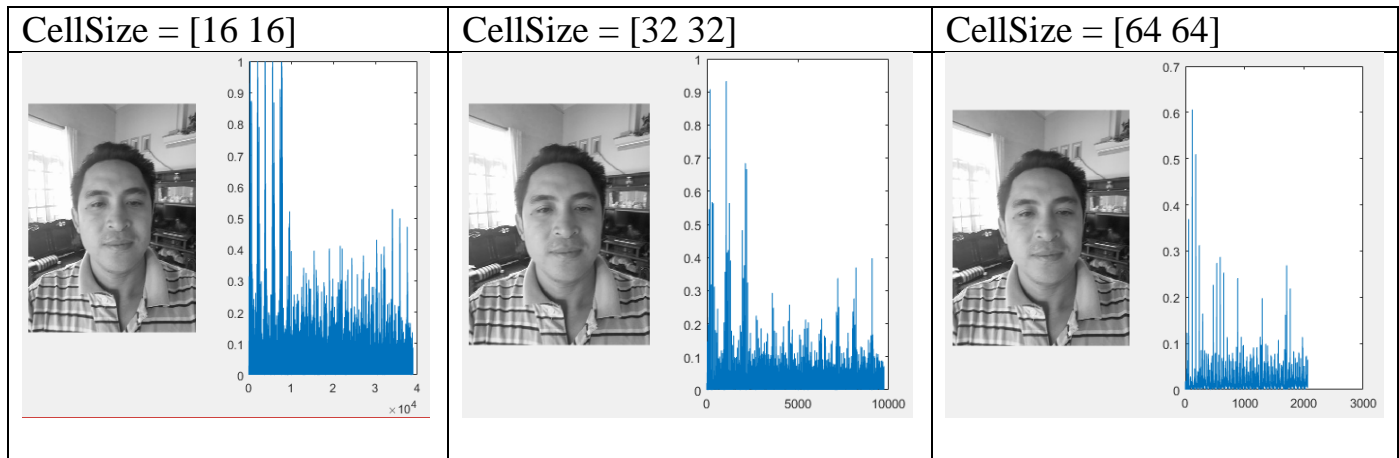
- ❖ Nạp dữ liệu train cho bài toán nhận dạng ảnh
- ❖ Lấy đặc trưng **LBP** cho từng ảnh trong dữ liệu train
- ❖ Xây dựng một model từ tập dữ liệu đặc trưng **LBP**
- ❖ Nạp dữ liệu test của bài toán nhận dạng ảnh
- ❖ Lấy đặc trưng **LBP** cho từng ảnh trong tập dữ liệu test.
- ❖ Sử dụng model để predict cho tập dữ liệu train

```

1. function LBPFeature()
2. %Read in a sample image and convert it to grayscale.

```

```
3. I = imread('TrongTri.jpg');
4. I = rgb2gray(I);
```



```
5.
6. %Extract unnormalized LBP features so that you can apply a custom
   normalization.
7. lbpFeatures = extractLBPFeatures(I,'CellSize',[64
   64],'Normalization','None');
8.
9. %Reshape the LBP features into a number of neighbors -by- number
10. % of cells array to access histograms for each individual cell.
11. numNeighbors = 8;
12. numBins = numNeighbors*(numNeighbors-1)+3;
13. lbpCellHists = reshape(lbpFeatures,numBins,[]);
14.
15. %Normalize each LBP cell histogram using L1 norm.
16. lbpCellHists = bsxfun(@divide,lbpCellHists,sum(lbpCellHists));
17.
18. %Reshape the LBP features vector back to 1-by- N feature vector.
19. lbpFeatures = reshape(lbpCellHists,1,[]);
20.
21. %Display the original image and the LBP features.
22. subplot(1,2,1);
23. imshow(I);
24. subplot(1,2,2);
25. plot(lbpFeatures);
26. end
```

Thay đổi cellsize ta lần lượt thu được các kết quả như sau:

2/ Create Bag of Visual Words (BoW)

- Nạp dữ liệu huấn luyện dữ liệu train của bài toán nhận dạng Fruits
- Xây dựng Bag-of-Word(BoW)
- Xây dựng mô hình phân lớp từ tập dữ liệu đặc trưng BoW.
- Phân lớp ảnh mới

Đánh giá kết quả phân lớp: đếm số lượng mẫu đúng và xuất kết quả.

```
1.function Bag_ofWords
2.%% Load Image Data Train
3.rootFolder = fullfile('DataTrain');
4.categories = {'Dua_hau', 'Dua_leo', 'Du_du', 'Thanh_long',
    'Ca_chua'};
5.imds = imageDatastore(fullfile(rootFolder, categories),
    'LabelSource', 'foldernames');
6.%%-----
7.tbl01 = countEachLabel(imds)
8.minSetCount = min(tbl01 {:, 2});
9.imds = splitEachLabel(imds, minSetCount, 'randomize');
10.    tbl02 = countEachLabel(imds)
11.    bag = bagOfFeatures(imds);
12.    img = readimage(imds,1);
13.    featureVector = encode(bag,img);
14.    figure
15.    bar(featureVector)
16.    title('Visual word occurrences')
17.    xlabel('Visual word index')
18.    ylabel('Frequency of occurrence')
19.    % -----
20.    categoryClassifier = trainImageCategoryClassifier(imds,
    bag);
21.    rootFolder = fullfile('DataTest');
22.    categories = {'Dua_leo', 'Du_du', 'Thanh_long', 'Ca_chua',
    'Dua_hau'};
23.    imds = imageDatastore(fullfile(rootFolder, categories),
    'LabelSource', 'foldernames');
24.    tbl01 = countEachLabel(imds)
25.    confMatrixTest = evaluate(categoryClassifier, imds);
26.    mean(diag(confMatrixTest));
27.    end
```

Phân tích và đánh giá kết quả mô hình dựa vào kết quả thực nghiệm

Kết quả của mô hình BoW cho bộ FruitsData

```
Command Window

Label      Count
-----
Ca_chua    50
Du_du      50
Dua_hau    50
Dua_leo    50
Thanh_long 50

Evaluating image category classifier for 5 categories.
-----

* Category 1: Ca_chua
* Category 2: Du_du
* Category 3: Dua_hau
* Category 4: Dua_leo
* Category 5: Thanh_long

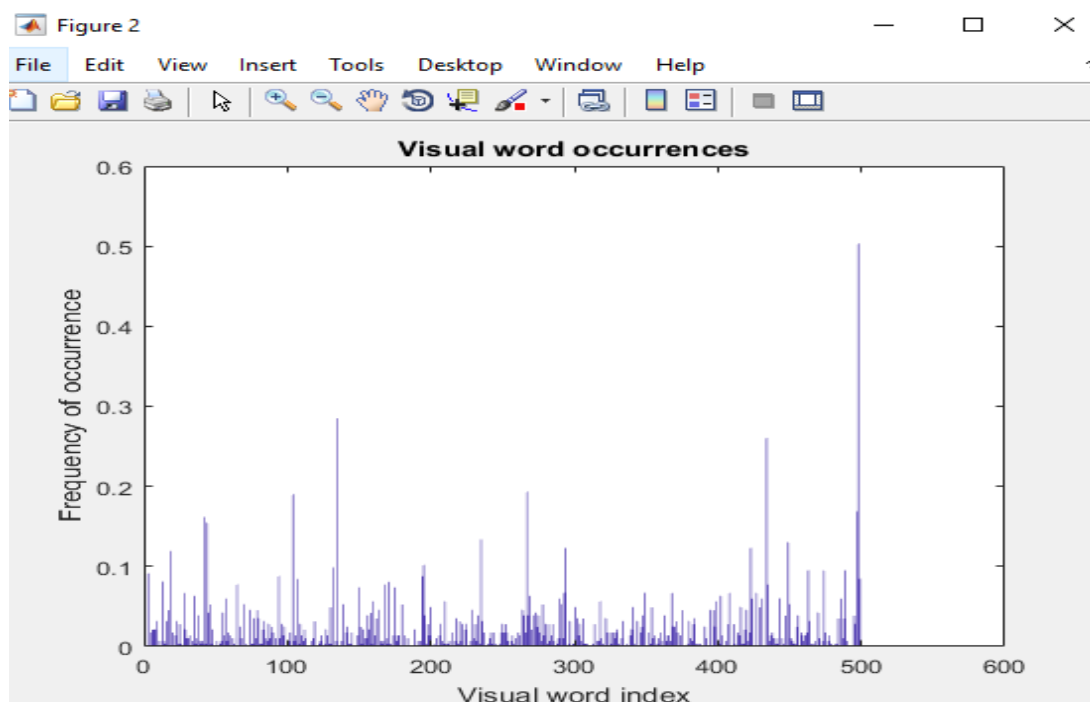
* Evaluating 250 images...done.

* Finished evaluating all the test sets.

* The confusion matrix for this test set is:
```

KNOWN	PREDICTED				
	Ca_chua	Du_du	Dua_hau	Dua_leo	Thanh_long
Ca_chua	0.88	0.02	0.00	0.00	0.10
Du_du	0.04	0.86	0.02	0.06	0.02
Dua_hau	0.04	0.02	0.90	0.02	0.02
Dua_leo	0.02	0.02	0.04	0.86	0.06
Thanh_long	0.10	0.02	0.04	0.00	0.84

* Average Accuracy is 0.87.



3/Machine learning methods: HoGFeatureKNN, HoGFeatureSVM

- Mô hình HOG: sử dụng rút trích đặc trưng Histogram of oriented gradients (HOG) và thuật toán K-nearest neighbor (k-NN)
- Mô hình HOG: sử dụng rút trích đặc trưng Histogram of oriented gradients (HOG) và thuật toán Support Vector Machine (SVM)

3.1/ HoGFeatureKNN:

- Nạp dữ liệu train của bài toán nhận dạng ảnh Fruits
- Lấy đặc trưng HOG cho từng ảnh trong dữ liệu train.
- Xây dựng một model từ tập dữ liệu đặc trưng HOG bằng thuật giải KNN.
- Nạp dữ liệu test của bài toán nhận dạng ảnh Fruits
- Sử dụng model để predict cho tập dữ liệu test.
- Đếm số lượng mẫu đúng và xuất kết quả.

a/ Thực nghiệm các loại rút trích đặc trưng trong Matlab (HOGFeatureKNN)

```
1. function HOGFeatureKNN()
2. %% Load Image Information from AnimalData Directory
3. faceDatabase = imageSet('FruitsData','recursive');
4. %% Split Database into Training & Test Sets 4680
5. [training,test] = partition(faceDatabase,[0.8 0.2]);
6. %% Extract HOG Features for training set
7. % 26244 = extractHOGFeatures(read(training(i),j)). L?u y?
8. trainingFeatures =
    zeros(size(training,2)*training(1).Count,26244);
9. TrainingfeatureCount = 1;
10.     for i=1:size(training,2)
11.         for j = 1:training(i).Count
12.             a. trainingFeatures(TrainingfeatureCount,:) =
                    extractHOGFeatures(read(training(i),j));
13.             b. trainingLabel{TrainingfeatureCount} =
                    training(i).Description;
14.             c. TrainingfeatureCount = TrainingfeatureCount + 1;
15.         end
16.         FruitstraingIndex{i} = training(i).Description;
17.     end
18. %% Create 40 class classifier using fitcecoc
19. MdKNN = fitcknn(trainingFeatures,trainingLabel);
20. %% Extract HOG Features for test set
21. % X = extractHOGFeatures(read(training(i),j)). Luy y: X =
    26244
22. testFeatures = zeros(size(test,2)*test(1).Count,26244);
23. TestfeatureCount = 1;
24.     for i=1:size(test,2)
```

```

22.     for j = 1:test(i).Count
    a. testFeatures(TestfeatureCount,:) =
        extractHOGFeatures(read(test(i),j));
    b. testLabel{TestfeatureCount} = test(i).Description;
    c. TestfeatureCount = TestfeatureCount + 1;
23.     end
24.     AnimaltestIndex{i} = test(i).Description;
25.     end
26.     %% Classify Test Images and count of number of sample
27.     predictedLabels = predict(MdKNN,testFeatures);
28.     lblPredicted = predictedLabels';
29.     nResult = strcmp(testLabel,lblPredicted);
30.     nCount = sum(nResult);
31.     nTest = size(testLabel,2);
32.     accuracy = nCount/nTest
33.     fprintf('\n So luong mau test :%d\n', nCount)
34.     fprintf('\n So luong mau dung :%d\n', nCount)
35.     fprintf('\n Ti le mau dung :%d\n', accuracy)
36.     %% Display five sample test images with their predicted
        labels.
37.     Fruits = 6;
38.     idx = [1 10 20 30 40];
39.     figure;
40.     for i = 1:numel(idx)
41.         label = predictedLabels(idx(i));
42.         lbl = label';
43.         subplot(2,3,i);imshow(read(test(i),Fruits));
44.         title(char(lbl));
45.     end
46.     end

```

3.2/ HOGFeatureSVM:

```

1. function HOGFeatureSVM()
2. %% Load Image Information from ATT AnimalData Directory
3. faceDatabase = imageSet('FruitsData','recursive');
4. %% Split Database into Training & Test Sets 4680
5. [training,test] = partition(faceDatabase,[0.8 0.2]);
6. %% Extract HOG Features for training set
7. % 26244 = extractHOGFeatures(read(training(i),j)). L?u y?
8. trainingFeatures =
    zeros(size(training,2)*training(1).Count,26244);
9. TrainingfeatureCount = 1;
10.    for i=1:size(training,2)
11.        for j = 1:training(i).Count
    a. trainingFeatures(TrainingfeatureCount,:) =
        extractHOGFeatures(read(training(i),j));
    b. trainingLabel{TrainingfeatureCount} =
        training(i).Description;
    c. TrainingfeatureCount = TrainingfeatureCount + 1;
12.        end
13.        FruitstraingIndex{i} = training(i).Description;
14.        end
15.        %% Create 40 class classifier using fitcecoc

```

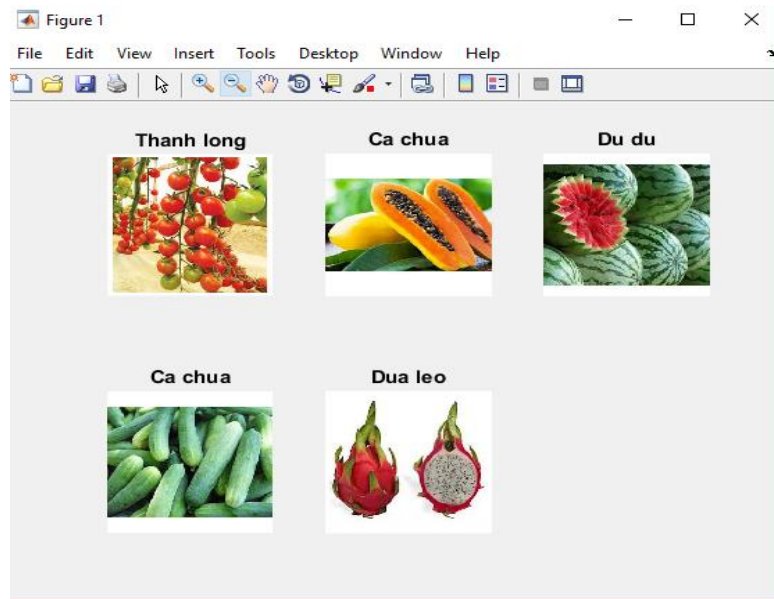
```

16.      MdSVM = fitcecoc(trainingFeatures,trainingLabel);

17.      %% Extract HOG Features for test set
18.      % X = extractHOGFeatures(read(training(i),j)). Luy y: X =
26244
19.      testFeatures = zeros(size(test,2)*test(1).Count,26244);
20.      TestfeatureCount = 1;
21.      for i=1:size(test,2)
22.          for j = 1:test(i).Count
23.              a. testFeatures(TestfeatureCount,:) =
                extractHOGFeatures(read(test(i),j));
                b. testLabel{TestfeatureCount} = test(i).Description;
                c. TestfeatureCount = TestfeatureCount + 1;
24.          end
25.          AnimaltestIndex{i} = test(i).Description;
26.      end
27.      %% Classify Test Images and count of number of sample
28.      predictedLabels = predict(MdSVM,testFeatures);
29.      lblPredicted = predictedLabels';
30.      nResult = strcmp(testLabel,lblPredicted);
31.      nCount = sum(nResult);
32.      nTest = size(testLabel,2);
33.      accuracy = nCount/nTest
34.      fprintf('\n So luong mau test :%d\n', nCount)
35.      fprintf('\n So luong mau dung :%d\n', nCount)
36.      fprintf('\n Ti le mau dung :%d\n', accuracy)
37.      %% Display five sample test images with their predicted
38.      labels.
39.      Fruits = 6;
40.      idx = [1 10 20 30 40];
41.      figure;
42.      for i = 1:numel(idx)
43.          label = predictedLabels(idx(i));
44.          lbl = label';
45.          subplot(2,3,i);imshow(read(test(i),Fruits));
46.          title(char(lbl));
47.      end
48.  end

```

Kết quả chạy thực nghiệm:



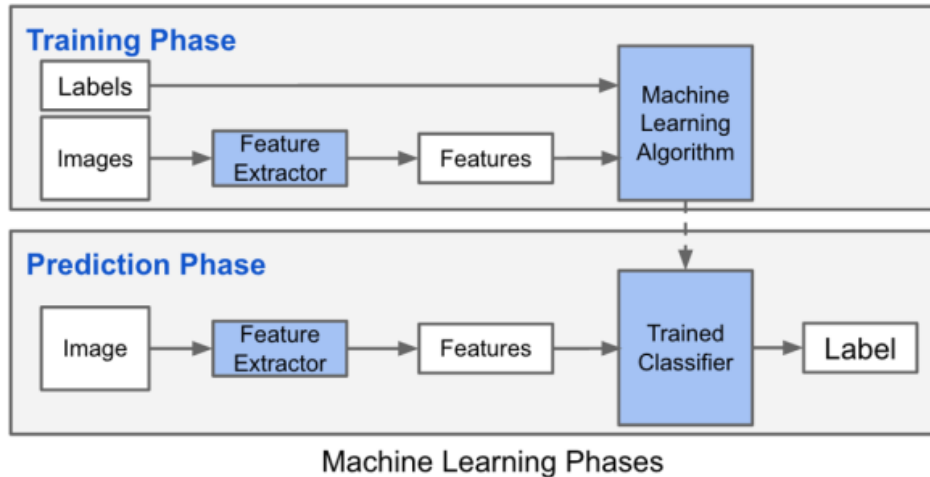
HOGFeaturneKNN	HOGFeaturneSVM
<pre>>> HOGFeaturneKNN accuracy = 0.4400 So luong mau test :22 So luong mau dung :22 Ti le mau dung :4.400000e-01</pre>	<pre>>> HOGFeaturneSVM accuracy = 0.6600 So luong mau test :33 So luong mau dung :33 Ti le mau dung :6.600000e-01</pre>

Thuật toán SVM cho kết quả tốt hơn KNN trên cùng bộ dữ liệu

4/ Machine learning methods: AlexNetFeatureKNN, AlexNetFeatureSVM

- Mô hình Alexnet: sử dụng rút trích đặc trưng Alexnet và thuật toán Support Vector Machine (SVM)
- Mô hình Alexnet: sử dụng rút trích đặc trưng Alexnet và thuật toán Support Vector Machine (k-NN)

Mô hình hệ thống:



Mô hình Alexnet: sử dụng rút trích đặc trưng Alexnet và thuật toán

Support Vector Machine (k-NN)

a/ Giới thiệu mô hình trong đồ án

- Nạp dữ liệu train của bài toán nhận dạng ảnh
- Nạp model alexnet (net)
- Sử dụng model(net) để trích chọn đặc trưng của dữ liệu datatrain.
- Xây dựng một model bằng cách sử dụng hàm **fitcknn** từ tập dữ liệu đặt trưng này.
- Nạp dữ liệu datatest của bài toán nhận dạng ảnh
- Sử dụng model (net) để trích chọn đặc trưng của dữ liệu datatest
- Sử dụng model để predict cho tập dữ liệu datatest
- Đếm số lượng mẫu đúng và xuất kết quả

b/ Thực nghiệm các loại rút trích đặc trưng trong Matlab

```

1.function AlexNetFeature()
2.%Load Data
3.images = imageDatastore('FruitsData', 'IncludeSubfolders',true,
    'LabelSource','foldernames');
4.%Split the data into 70% training and 30% test data.
5.[trainingImages,testImages] =
    splitEachLabel(images,0.7,'randomized');
6.%Load Pretrained Network (Extract Feature using AlexNet network)
7.net = alexnet;
8.%Display the network architecture. The network has five
    convolutional layers and three fully connected layers.
9.net.Layers

```

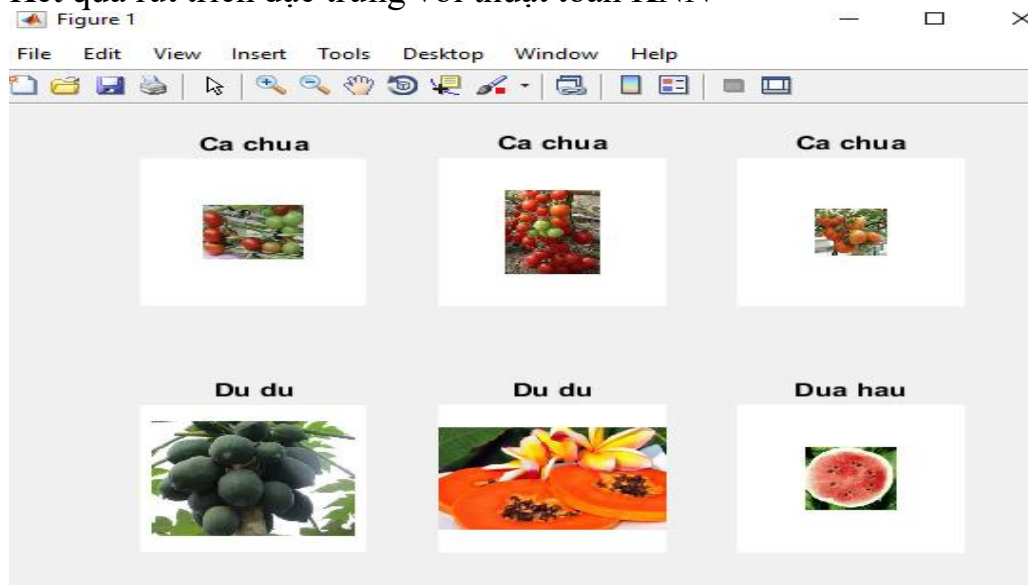
```

10. %Extract Image Features
11. layer = 'fc7';
12. trainingFeatures = activations(net,trainingImages,layer);
13. testFeatures = activations(net,testImages,layer);

14. %Extract the class labels from the training and test data.
15. trainingLabels = trainingImages.Labels;
16. testLabels = testImages.Labels;
17. %Fit Image Classifier
18. classifier = fitcecoc(trainingFeatures,trainingLabels);
19. %classifier = fitcknn(trainingFeatures,trainingLabels);
20. %Classify Test Images
21. predictedLabels = predict(classifier,testFeatures);
22. %Display four sample test images with their predicted
    labels.
23. idx = [1 7 14 21 28 35];
24. figure
25. for i = 1:numel(idx)
26.     subplot(2,3,i)
27.     I = readimage(testImages,idx(i));
28.     label = predictedLabels(idx(i));
29.     imshow(I)
30.     title(char(label))
31. end
32. %Calculate the classification accuracy on the test set.
    Accuracy is the fraction of labels that the network predicts
    correctly.
33. accuracy = mean(predictedLabels == testLabels)
34. end

```

Kết quả rút trích đặc trưng với thuật toán KNN




```

Command Window

4 'norm1' Cross Channel Normalization cross channel normalization with 5 channels per element
5 'pool1' Max Pooling 3x3 max pooling with stride [2 2] and padding [0 0]
6 'conv2' Convolution 256 5x5x48 convolutions with stride [1 1] and padding [2 2]
7 'relu2' ReLU ReLU
8 'norm2' Cross Channel Normalization cross channel normalization with 5 channels per element
9 'pool2' Max Pooling 3x3 max pooling with stride [2 2] and padding [0 0]
10 'conv3' Convolution 384 3x3x256 convolutions with stride [1 1] and padding [1 1]
11 'relu3' ReLU ReLU
12 'conv4' Convolution 384 3x3x192 convolutions with stride [1 1] and padding [1 1]
13 'relu4' ReLU ReLU
14 'conv5' Convolution 256 3x3x192 convolutions with stride [1 1] and padding [1 1]
15 'relu5' ReLU ReLU
16 'pool5' Max Pooling 3x3 max pooling with stride [2 2] and padding [0 0]
17 'fc6' Fully Connected 4096 fully connected layer
18 'relu6' ReLU ReLU
19 'drop6' Dropout 50% dropout
20 'fc7' Fully Connected 4096 fully connected layer
21 'relu7' ReLU ReLU
22 'drop7' Dropout 50% dropout
23 'fc8' Fully Connected 1000 fully connected layer
24 'prob' Softmax softmax
25 'output' Classification Output crossentropyx with 'tench', 'goldfish', and 998 other classes

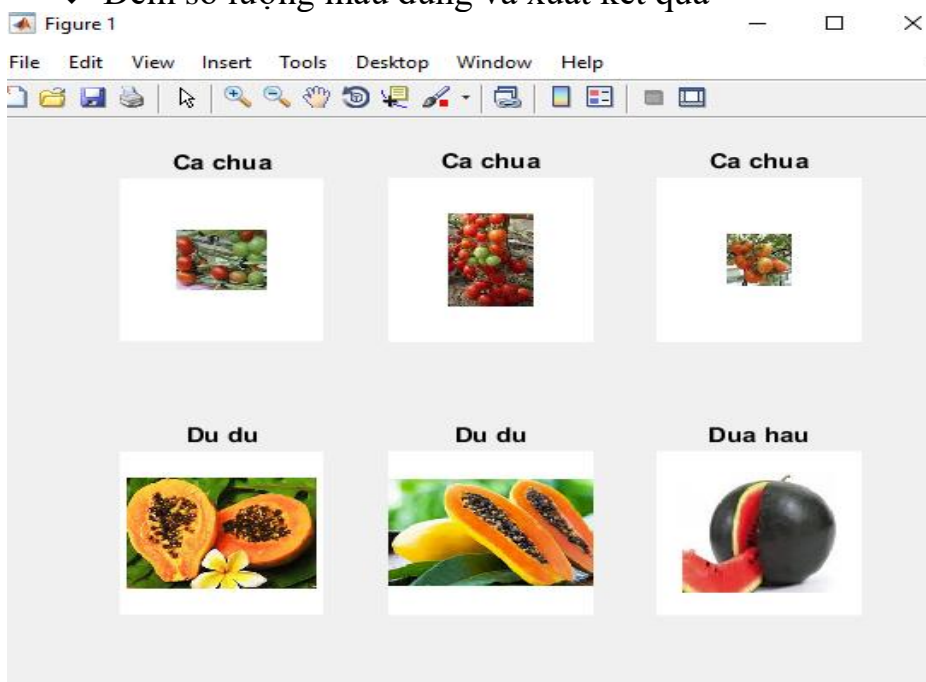
accuracy =

0.8800

```

Kết quả rút trích đặc trưng với thuật toán SVM

- ❖ Nạp dữ liệu train của bài toán nhận dạng ảnh FruitsData
- ❖ Nạp model alexnet (net)
- ❖ Sử dụng model(net) để trích chọn đặc trưng của dữ liệu datatrain.
- ❖ Xây dựng một model bằng cách sử dụng hàm fitcecoc từ tập dữ liệu đặt trưng này.
- ❖ Nạp dữ liệu datatest của bài toán nhận dạng ảnh FruitsData
- ❖ Sử dụng model (net) để trích chọn đặc trưng của dữ liệu datatest
- ❖ Sử dụng model để predict cho tập dữ liệu test
- ❖ Đếm số lượng mẫu đúng và xuất kết quả



```

ommand Window
4 'norm1' Cross Channel Normalization cross channel normalization with 5 channels per element
5 'pool1' Max Pooling 3x3 max pooling with stride [2 2] and padding [0 0]
6 'conv2' Convolution 256 5x5x48 convolutions with stride [1 1] and padding [2 2]
7 'relu2' ReLU ReLU
8 'norm2' Cross Channel Normalization cross channel normalization with 5 channels per element
9 'pool2' Max Pooling 3x3 max pooling with stride [2 2] and padding [0 0]
10 'conv3' Convolution 384 3x3x256 convolutions with stride [1 1] and padding [1 1]
11 'relu3' ReLU ReLU
12 'conv4' Convolution 384 3x3x192 convolutions with stride [1 1] and padding [1 1]
13 'relu4' ReLU ReLU
14 'conv5' Convolution 256 3x3x192 convolutions with stride [1 1] and padding [1 1]
15 'relu5' ReLU ReLU
16 'pool5' Max Pooling 3x3 max pooling with stride [2 2] and padding [0 0]
17 'fc6' Fully Connected 4096 fully connected layer
18 'relu6' ReLU ReLU
19 'drop6' Dropout 50% dropout
20 'fc7' Fully Connected 4096 fully connected layer
21 'relu7' ReLU ReLU
22 'drop7' Dropout 50% dropout
23 'fc8' Fully Connected 1000 fully connected layer
24 'prob' Softmax softmax
25 'output' Classification Output crossentropyex with 'tench', 'goldfish', and 998 other classes

accuracy =

0.9200

```

So sánh giữa hai thuật toán trên độ chính xác

AlexNetFeatureKNN	AlexNetFeatureSVM
88%	92%

3.3. So sánh các features và thuật toán cho mô hình phân loại ảnh

Features	HOG	HOG	BoW	AlexNet	AlexNet
Algorithm	KNN	SVM	B- SVM	SVM	KNN
Classifier	5	5	5	5	5
Data train	250	250	250	250	250
Data test	50	50	50	50	50
Accuracy	44%	66%	90%	92%	88%

Bảng so sánh kết quả của mô hình

Trên đây là bảng thống kê kết quả đạt được, kết quả này chỉ mang tính chất tham khảo vì bộ dữ liệu thực nghiệm quá ít mẫu nên cũng không thể hiện hết được tính chính xác. Trong tương lai nếu có điều kiện về thiết bị nghiên cứu chúng tôi sẽ thử nghiệm trên bộ dữ liệu lớn hơn.

III/Kết luận

Rút trích đặc trưng: đối với ảnh màu có độ phân giải ngày càng cao nên trong phần tùy biến các thông số chúng ta nên chọn cellsize lớn hơn.

Thuật toán: k-NN, SVM, Machine Learning thì thuật toán Machine Learning cho kết quả chính xác nhất. Tuy nhiên, gặp khó khăn trong vấn đề cài đặt thêm công cụ (Neural Network Toolbox Model for AlexNet Network) và áp dụng vào bộ dữ liệu mới do chúng ta tạo ra. Thời gian chạy thuật toán: cùng một bộ dữ liệu thuật toán ML chạy nhanh hơn (theo định tính).

IV/Tài liệu tham khảo

1. Bài giảng và thực hành Nhận dạng thị giác và ứng dụng của Ts. Lê Đình Duy và Ts. Nguyễn Tấn Trần Minh Khang
2. <https://www.mathworks.com/help/vision/ref/extracthogfeatures.html>
3. <https://www.mathworks.com/help/vision/ref/extractlbpfeatures.html>
4. <https://www.mathworks.com/help/vision/ref/bagoffeatures-class.html>
5. <https://www.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html>
6. <https://www.mathworks.com/examples/matlab/community/22754-deep-learning-example-feature-extraction-using-alexnet-and-cifar-10-dataset>