Object Recognition and Localization

Badri Patro & Ganesh Boddupally

EE698M: Project Presentation

Guidance: Prof. Tanaya Guha

Indian Institute of Technology, Kanpur

April 25, 2016

Problem Statement

- This presentation will address Object Recognition and Localization in a video.
- Detect and recognize a particular object in a video and then find out corresponding timing details of that object present in the video sequence, i.e, what are frames that object is available or the different time interval this object is available in the video scene.

Problem Statement

- This presentation will address Object Recognition and Localization in a video.
- Detect and recognize a particular object in a video and then find out corresponding timing details of that object present in the video sequence, i.e, what are frames that object is available or the different time interval this object is available in the video scene.

Problem Statement

- This presentation will address Object Recognition and Localization in a video.
- Detect and recognize a particular object in a video and then find out corresponding timing details of that object present in the video sequence, i.e, what are frames that object is available or the different time interval this object is available in the video scene.

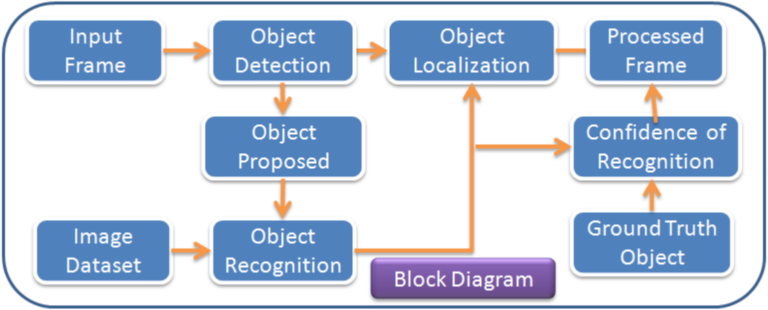
Problem Statement

- This presentation will address Object Recognition and Localization in a video.
- Detect and recognize a particular object in a video and then find out corresponding timing details of that object present in the video sequence, i.e, what are frames that object is available or the different time interval this object is available in the video scene.

Problem Statement

- This presentation will address Object Recognition and Localization in a video.
- Detect and recognize a particular object in a video and then find out corresponding timing details of that object present in the video sequence, i.e, what are frames that object is available or the different time interval this object is available in the video scene.

Block diagram of Proposed Algorithm:



Proposed Algorithm:

- Object Detection
- Object Recognition
- Object Localization

Proposed Algorithm:

- Object Detection
- Object Recognition
- Object Localization

Proposed Algorithm:

- Object Detection
- Object Recognition
- Object Localization

Object Detection methods:

- Frame Difference
- Mean Filter
- Running Gaussian
- GMM

Object Detection methods:

- Frame Difference
- Mean Filter
- Running Gaussian
- GMM

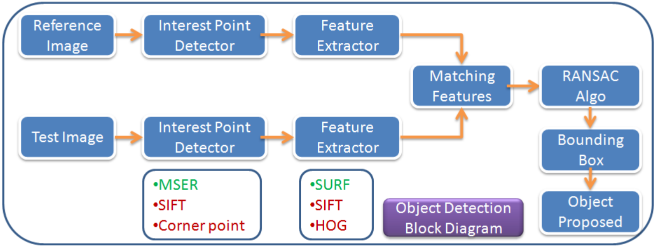
Object Detection methods:

- Frame Difference
- Mean Filter
- Running Gaussian
- GMM

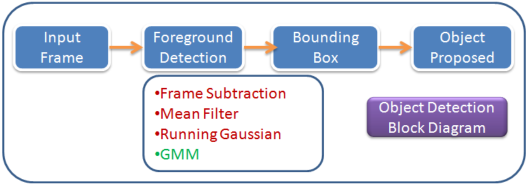
Object Detection methods:

- Frame Difference
- Mean Filter
- Running Gaussian
- GMM

Object Detection Method 1:



Object Detection Method 2:



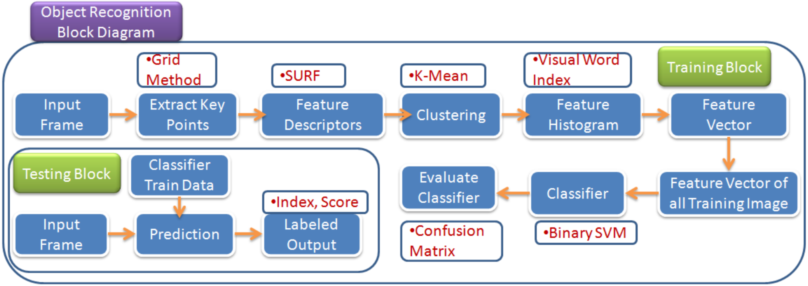
Block Diagram of Object Recognition :

- Object Detection
- Object Recognition

Block Diagram of Object Recognition :

- Object Detection
- Object Recognition

Approached Methods with block Diagrams and Demo:



Training:

- Extract Key Points locations using the Grid method.
  - GridStep is [8 8] and
  - BlockWidth is [32 64 96 128] :to take care of Scale Information
- Feature Descriptors: Extracting SURF features from the selected interest point.
  - Strongest Features: 80 percent of the strongest features.
  - Find the minimum no of strongest feature among all the data set.
  - Lets say M is minimum no of feature among all N(14) Image data set, each is having 100,200, 300 images.

Training:

- Extract Key Points locations using the Grid method.
  - GridStep is [8 8] and
  - BlockWidth is [32 64 96 128] :to take care of Scale Information
- Feature Descriptors: Extracting SURF features from the selected interest point.
  - Strongest Features: 80 percent of the strongest features.
  - Find the minimum no of strongest feature among all the data set.
  - Lets say M is minimum no of feature among all N(14) Image data set, each is having 100,200, 300 images.

Training:

- Extract Key Points locations using the Grid method.
  - GridStep is [8 8] and
  - BlockWidth is [32 64 96 128] :to take care of Scale Information
- Feature Descriptors: Extracting SURF features from the selected interest point.
  - Strongest Features: 80 percent of the strongest features.
  - Find the minimum no of strongest feature among all the data set.
  - Lets say M is minimum no of feature among all N(14) Image data set, each is having 100,200, 300 images.

Training:

- Extract Key Points locations using the Grid method.
  - GridStep is [8 8] and
  - BlockWidth is [32 64 96 128] :to take care of Scale Information
- Feature Descriptors: Extracting SURF features from the selected interest point.
  - Strongest Features: 80 percent of the strongest features.
  - Find the minimum no of strongest feature among all the data set.
  - Lets say M is minimum no of feature among all N(14) Image data set, each is having 100,200, 300 images.

Training:

- Extract Key Points locations using the Grid method.
  - GridStep is [8 8] and
  - BlockWidth is [32 64 96 128] :to take care of Scale Information
- Feature Descriptors: Extracting SURF features from the selected interest point.
  - Strongest Features: 80 percent of the strongest features.
  - Find the minimum no of strongest feature among all the data set.
  - Lets say M is minimum no of feature among all N(14) Image data set, each is having 100,200, 300 images.

Training:

- Extract Key Points locations using the Grid method.
  - GridStep is [8 8] and
  - BlockWidth is [32 64 96 128] :to take care of Scale Information
- Feature Descriptors: Extracting SURF features from the selected interest point.
  - Strongest Features: 80 percent of the strongest features.
  - Find the minimum no of strongest feature among all the data set.
  - Lets say M is minimum no of feature among all N(14) Image data set, each is having 100,200, 300 images.

Training:

- Extract Key Points locations using the Grid method.
  - GridStep is [8 8] and
  - BlockWidth is [32 64 96 128] :to take care of Scale Information
- Feature Descriptors: Extracting SURF features from the selected interest point.
  - Strongest Features: 80 percent of the strongest features.
  - Find the minimum no of strongest feature among all the data set.
  - Lets say M is minimum no of feature among all N(14) Image data set, each is having 100,200, 300 images.

Training:

- Clustering: Approximate Nearest Neighbor is used to cluster all the feature .
  - Divide Complete features into K(500) visual vocabulary words.
    - Number of clusters (K): 500.
    - Number of features : M* N;
    - Initialization the cluster centers
    - termination criteria: 100 time loop or cluster distance error ¡ threshold.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Feature Vector: generate feature vector corresponding to each image

Training:

- Clustering: Approximate Nearest Neighbor is used to cluster all the feature .
  - Divide Complete features into K(500) visual vocabulary words.
    - Number of clusters (K): 500.
    - Number of features : M* N;
    - Initialization the cluster centers
    - termination criteria: 100 time loop or cluster distance error ¡ threshold.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Feature Vector: generate feature vector corresponding to each image

Training:

- Clustering: Approximate Nearest Neighbor is used to cluster all the feature .
  - Divide Complete features into K(500) visual vocabulary words.
    - Number of clusters (K): 500.
    - Number of features : M* N;
    - Initialization the cluster centers
    - termination criteria: 100 time loop or cluster distance error ¡ threshold.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Feature Vector: generate feature vector corresponding to each image

Training:

- Clustering: Approximate Nearest Neighbor is used to cluster all the feature .
  - Divide Complete features into K(500) visual vocabulary words.
    - Number of clusters (K): 500.
    - Number of features : M* N;
    - Initialization the cluster centers
    - termination criteria: 100 time loop or cluster distance error ¡ threshold.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Feature Vector: generate feature vector corresponding to each image

Training:

- Clustering: Approximate Nearest Neighbor is used to cluster all the feature .
  - Divide Complete features into K(500) visual vocabulary words.
    - Number of clusters (K): 500.
    - Number of features : M* N;
    - Initialization the cluster centers
    - termination criteria: 100 time loop or cluster distance error ¡ threshold.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Feature Vector: generate feature vector corresponding to each image

Training:

- Clustering: Approximate Nearest Neighbor is used to cluster all the feature .
  - Divide Complete features into K(500) visual vocabulary words.
    - Number of clusters (K): 500.
    - Number of features : M* N;
    - Initialization the cluster centers
    - termination criteria: 100 time loop or cluster distance error ¡ threshold.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Feature Vector: generate feature vector corresponding to each image

Training:

- Clustering: Approximate Nearest Neighbor is used to cluster all the feature .
  - Divide Complete features into K(500) visual vocabulary words.
    - Number of clusters (K): 500.
    - Number of features : M* N;
    - Initialization the cluster centers
    - termination criteria: 100 time loop or cluster distance error ¡ threshold.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Feature Vector: generate feature vector corresponding to each image

Training:

- Clustering: Approximate Nearest Neighbor is used to cluster all the feature .
  - Divide Complete features into K(500) visual vocabulary words.
    - Number of clusters (K): 500.
    - Number of features : M* N;
    - Initialization the cluster centers
    - termination criteria: 100 time loop or cluster distance error ¡ threshold.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Feature Vector: generate feature vector corresponding to each image

Training:

- Clustering: Approximate Nearest Neighbor is used to cluster all the feature .
  - Divide Complete features into K(500) visual vocabulary words.
    - Number of clusters (K): 500.
    - Number of features : M* N;
    - Initialization the cluster centers
    - termination criteria: 100 time loop or cluster distance error ¡ threshold.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Feature Vector: generate feature vector corresponding to each image

Training:

- Clustering: Approximate Nearest Neighbor is used to cluster all the feature .
  - Divide Complete features into K(500) visual vocabulary words.
    - Number of clusters (K): 500.
    - Number of features : M* N;
    - Initialization the cluster centers
    - termination criteria: 100 time loop or cluster distance error ¡ threshold.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Feature Vector: generate feature vector corresponding to each image

Training:

- Clustering: Approximate Nearest Neighbor is used to cluster all the feature .
  - Divide Complete features into K(500) visual vocabulary words.
    - Number of clusters (K): 500.
    - Number of features : M* N;
    - Initialization the cluster centers
    - termination criteria: 100 time loop or cluster distance error ¡ threshold.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Feature Vector: generate feature vector corresponding to each image

Training:

- Training data : Repeat privious two slide for each image in the training set to create the training data
  - Generate 101 feature histogram for car training image.
  - similarly generate feature histogram all training image.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Classifier:
  - Encoded training images from each category are fed into a classifier training process.
  - The function trains a multiclass classifier using the error-correcting output codes (ECOC) framework with the help of binary support vector machine (SVM) classifiers

Training:

- Training data : Repeat privious two slide for each image in the training set to create the training data
  - Generate 101 feature histogram for car training image.
  - similarly generate feature histogram all training image.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Classifier:
  - Encoded training images from each category are fed into a classifier training process.
  - The function trains a multiclass classifier using the error-correcting output codes (ECOC) framework with the help of binary support vector machine (SVM) classifiers

Training:

- Training data : Repeat privious two slide for each image in the training set to create the training data
  - Generate 101 feature histogram for car training image.
  - similarly generate feature histogram all training image.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Classifier:
  - Encoded training images from each category are fed into a classifier training process.
  - The function trains a multiclass classifier using the error-correcting output codes (ECOC) framework with the help of binary support vector machine (SVM) classifiers

Training:

- Training data : Repeat privious two slide for each image in the training set to create the training data
  - Generate 101 feature histogram for car training image.
  - similarly generate feature histogram all training image.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Classifier:
  - Encoded training images from each category are fed into a classifier training process.
  - The function trains a multiclass classifier using the error-correcting output codes (ECOC) framework with the help of binary support vector machine (SVM) classifiers

Training:

- Training data : Repeat privious two slide for each image in the training set to create the training data
  - Generate 101 feature histogram for car training image.
  - similarly generate feature histogram all training image.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Classifier:
  - Encoded training images from each category are fed into a classifier training process.
  - The function trains a multiclass classifier using the error-correcting output codes (ECOC) framework with the help of binary support vector machine (SVM) classifiers

Training:

- Training data : Repeat privious two slide for each image in the training set to create the training data
  - Generate 101 feature histogram for car training image.
  - similarly generate feature histogram all training image.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Classifier:
  - Encoded training images from each category are fed into a classifier training process.
  - The function trains a multiclass classifier using the error-correcting output codes (ECOC) framework with the help of binary support vector machine (SVM) classifiers

Training:

- Training data : Repeat privious two slide for each image in the training set to create the training data
  - Generate 101 feature histogram for car training image.
  - similarly generate feature histogram all training image.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Classifier:
  - Encoded training images from each category are fed into a classifier training process.
  - The function trains a multiclass classifier using the error-correcting output codes (ECOC) framework with the help of binary support vector machine (SVM) classifiers

Training:

- Training data : Repeat privious two slide for each image in the training set to create the training data
  - Generate 101 feature histogram for car training image.
  - similarly generate feature histogram all training image.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Classifier:
  - Encoded training images from each category are fed into a classifier training process.
  - The function trains a multiclass classifier using the error-correcting output codes (ECOC) framework with the help of binary support vector machine (SVM) classifiers

Training:

- Training data : Repeat privious two slide for each image in the training set to create the training data
  - Generate 101 feature histogram for car training image.
  - similarly generate feature histogram all training image.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Classifier:
  - Encoded training images from each category are fed into a classifier training process.
  - The function trains a multiclass classifier using the error-correcting output codes (ECOC) framework with the help of binary support vector machine (SVM) classifiers

Training:

- Training data : Repeat privious two slide for each image in the training set to create the training data
  - Generate 101 feature histogram for car training image.
  - similarly generate feature histogram all training image.
- Feature Histogram:
  - Generate No of word count present in each cluster
  - per each image find how many word are present in 500 cluster.
  - each cluster represent as visual word index(500 visual index)
- Classifier:
  - Encoded training images from each category are fed into a classifier training process.
  - The function trains a multiclass classifier using the error-correcting output codes (ECOC) framework with the help of binary support vector machine (SVM) classifiers

Results:

```
ObjectRecognitionTrainingTesting
---Start of TrainingObjectClassifier function---

object_description =

  Columns 1 through 6

    'car_side'    'Leopards'    'ant'    'butterfly'    'Motorbikes'    'ferry'

  Columns 7 through 13

    'cup'    'elephant'    'panda'    'cub'    'car'    'tiger_cartoon'    'rhino'

  Column 14

    'lion'


object_count =

   123   200    42    91   798    67    57    64    38    32   101    40    59    40


object_count =

    32    32    32    32    32    32    32    32    32    32    32    32    32    32
```

Results:

```
Creating Bag-Of-Features from 14 image sets.
--------------------------------------------
* Image set 1: car_side.
* Image set 2: Leopards.
* Image set 3: ant.
* Image set 4: butterfly.
* Image set 5: Motorbikes.
* Image set 6: ferry.
* Image set 7: cup.
* Image set 8: elephant.
* Image set 9: panda.
* Image set 10: cub.
* Image set 11: car.
* Image set 12: tiger_cartoon.
* Image set 13: rhino.
* Image set 14: lion.

* Extracting SURF features using the Grid selection method.
** The GridStep is [8 8] and the BlockWidth is [32 64 96 128].

* Extracting features from 10 images in image set 1...done. Extracted 38000 features.
* Extracting features from 10 images in image set 2...done. Extracted 15360 features.
* Extracting features from 10 images in image set 3...done. Extracted 40128 features.
* Extracting features from 10 images in image set 4...done. Extracted 44688 features.
* Extracting features from 10 images in image set 5...done. Extracted 25552 features.
* Extracting features from 10 images in image set 6...done. Extracted 33744 features.
* Extracting features from 10 images in image set 7...done. Extracted 48336 features.
* Extracting features from 10 images in image set 8...done. Extracted 44536 features.
* Extracting features from 10 images in image set 9...done. Extracted 47272 features.
* Extracting features from 10 images in image set 10...done. Extracted 32060 features.
* Extracting features from 10 images in image set 11...done. Extracted 13824 features.
* Extracting features from 10 images in image set 12...done. Extracted 32760 features.
* Extracting features from 10 images in image set 13...done. Extracted 41800 features.
* Extracting features from 10 images in image set 14...done. Extracted 17732 features.
```

Results:

*F:\Course_Work_PhD\Course_Work_IITK\MATLAB_Workspace\Video_Processing_EE608A\EE698M_Course_Projects\Complete_BK\Computer_Vision_Project\11_04_16\object_recognition_tra

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

Assignment_3_Report.tex | Assignment_2_Report.tex | CreateObjects.m | Assignment_1_EE627A.tex | CS676A_Project_Proposal.tex | EE698M_Paper_Presentation.tex

```
* Extracting features from 10 images in image set 6...done. Extracted 33744 features.
* Extracting features from 10 images in image set 7...done. Extracted 48336 features.
* Extracting features from 10 images in image set 8...done. Extracted 44536 features.
* Extracting features from 10 images in image set 9...done. Extracted 47272 features.
* Extracting features from 10 images in image set 10...done. Extracted 32060 features.
* Extracting features from 10 images in image set 11...done. Extracted 13824 features.
* Extracting features from 10 images in image set 12...done. Extracted 32760 features.
* Extracting features from 10 images in image set 13...done. Extracted 41800 features.
* Extracting features from 10 images in image set 14...done. Extracted 17732 features.

* Keeping 80 percent of the strongest features from each image set.

* Balancing the number of features across all image sets to improve clustering.
** Image set 11 has the least number of strongest features: 11059.
** Using the strongest 11059 features from each of the other image sets.

* Using K-Means clustering to create a 500 word visual vocabulary.
* Number of features          : 154826
* Number of clusters (K)      : 500

* Initializing cluster centers...0.40%0.60%0.80%1.00%1.20%1.40%1.60%1.80%2.00%2.20%2.40%2.60%2.80%3.00%3.20%3.40%3.60%3.
4.80%5.00%5.20%5.40%5.60%5.80%6.00%6.20%6.40%6.60%6.80%7.00%7.20%7.40%7.60%7.80%8.00%8.20%8.40%8.60%8.80%9.00%9.20%9.40%
%37.20%37.40%37.60%37.80%38.00%38.20%38.40%38.60%38.80%39.00%39.20%39.40%39.60%39.80%40.00%40.20%40.40%40.60%40.80%41.00%
%42.40%42.60%42.80%43.00%43.20%43.40%43.60%43.80%44.00%44.20%44.40%44.60%44.800%57.00%57.20%57.40%57.60%57.80%58.00%58.2
%76.20%76.40%76.60%76.80%77.00%77.20%77.40%77.60%77.80%78.00%78.20%78.40%78.60%78.80%79.00%79.20%79.40%79.60%79.80%80.00%
%97.20%97.40%97.60%97.80%98.00%98.20%98.40%98.60%98.80%99.00%99.20%99.40%99.60%99.80%100.00%.


* Clustering...completed 0/100 iterations completed 1/100 iterations (~0.71 seconds/iteration)completed 2/100 iterations
3/100 iterations (~0.57 seconds/iteration)completed 4/100 iterations (~0.73 seconds/iteration)completed 5/100 iterations
30/100 iterations (~0.70 seconds/iteration)completed 31/100 iterations (~0.59 seconds/iteration)completed 32/100 iterati
33/100 iterations (~0.58 seconds/iteration)completed 34/100 iterations (~0.56 seconds/iteration)completed 35/100 iterati
36/100 iterations (~0.70 seconds/iteration)...converged in 36 iterations.

* Finished creating Bag-Of-Features
```

Normal text file   length : 13332 lines : 257   Ln : 88 Col : 6 Sel : 0 | 0

Results:

```
Training an image category classifier for 14
--------------------------------------------
* Category 1: car_side
* Category 2: Leopards
* Category 3: ant
* Category 4: butterfly
* Category 5: Motorbikes
* Category 6: ferry
* Category 7: cup
* Category 8: elephant
* Category 9: panda
* Category 10: cub
* Category 11: car
* Category 12: tiger_cartoon
* Category 13: rhino
* Category 14: lion

* Encoding features for category 1...done.
* Encoding features for category 2...done.
* Encoding features for category 3...done.
* Encoding features for category 4...done.
* Encoding features for category 5...done.
* Encoding features for category 6...done.
* Encoding features for category 7...done.
* Encoding features for category 8...done.
* Encoding features for category 9...done.
* Encoding features for category 10...done.
* Encoding features for category 11...done.
* Encoding features for category 12...done.
* Encoding features for category 13...done.
* Encoding features for category 14...done.

* Finished training the category classifier.
```

Results:

* The confusion matrix for Training set is:

PREDICTED

KNOWN	car_side	Leopards	ant	butterfly	Motorbikes	ferry	cup	elephant	panda	cub	car	tiger_cartoon	rhino	lion
car_side	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Leopards	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ant	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
butterfly	0.00	0.10	0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00
Motorbikes	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ferry	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
cup	0.00	0.00	0.00	0.00	0.10	0.00	0.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00
elephant	0.10	0.10	0.00	0.00	0.00	0.00	0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.00
panda	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
cub	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
car	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
tiger_cartoon	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
rhino	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.90	0.00
lion	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.90

* Average Accuracy is 0.95.

confusion_matrix_training_set

Results:

```
Evaluating image category classifier for 14 categori
------------------------------------------------------

* Category 1: car_side
* Category 2: Leopards
* Category 3: ant
* Category 4: butterfly
* Category 5: Motorbikes
* Category 6: ferry
* Category 7: cup
* Category 8: elephant
* Category 9: panda
* Category 10: cub
* Category 11: car
* Category 12: tiger_cartoon
* Category 13: rhino
* Category 14: lion

* Evaluating 22 images from category 1...done.
* Evaluating 22 images from category 2...done.
* Evaluating 22 images from category 3...done.
* Evaluating 22 images from category 4...done.
* Evaluating 22 images from category 5...done.
* Evaluating 22 images from category 6...done.
* Evaluating 22 images from category 7...done.
* Evaluating 22 images from category 8...done.
* Evaluating 22 images from category 9...done.
* Evaluating 22 images from category 10...done.
* Evaluating 22 images from category 11...done.
* Evaluating 22 images from category 12...done.
* Evaluating 22 images from category 13...done.
* Evaluating 22 images from category 14...done.

* Finished evaluating all the test sets.

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

Results:

* The confusion matrix for this test set is:

PREDICTED

KNOWN	car_side	Leopards	ant	butterfly	Motorbikes	ferry	cup	elephant	panda	cub	car	tiger_cartoon	rhino	lion
car_side	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Leopards	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ant	0.05	0.00	0.05	0.41	0.05	0.00	0.05	0.05	0.14	0.05	0.00	0.00	0.05	0.14
butterfly	0.00	0.00	0.09	0.27	0.14	0.00	0.09	0.05	0.14	0.05	0.00	0.00	0.05	0.14
Motorbikes	0.00	0.00	0.05	0.00	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ferry	0.32	0.00	0.00	0.00	0.09	0.41	0.00	0.00	0.00	0.00	0.18	0.00	0.00	0.00
cup	0.23	0.00	0.00	0.09	0.14	0.00	0.32	0.00	0.14	0.00	0.05	0.00	0.05	0.00
elephant	0.09	0.00	0.00	0.05	0.05	0.05	0.00	0.23	0.05	0.14	0.00	0.05	0.18	0.14
panda	0.00	0.00	0.00	0.23	0.18	0.00	0.00	0.09	0.32	0.05	0.00	0.00	0.05	0.09
cub	0.05	0.14	0.05	0.00	0.00	0.00	0.00	0.05	0.00	0.41	0.00	0.09	0.05	0.18
car	0.18	0.00	0.00	0.00	0.00	0.09	0.00	0.00	0.05	0.00	0.68	0.00	0.00	0.00
tiger_cartoon	0.00	0.14	0.00	0.00	0.00	0.05	0.14	0.00	0.00	0.05	0.00	0.50	0.00	0.14
rhino	0.05	0.14	0.00	0.05	0.00	0.05	0.05	0.05	0.05	0.14	0.00	0.05	0.32	0.09
lion	0.00	0.05	0.05	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.09	0.00	0.77

* Average Accuracy is 0.52.

Approches:

- Object proposed
- Test Object Classifier
- Bounding Box

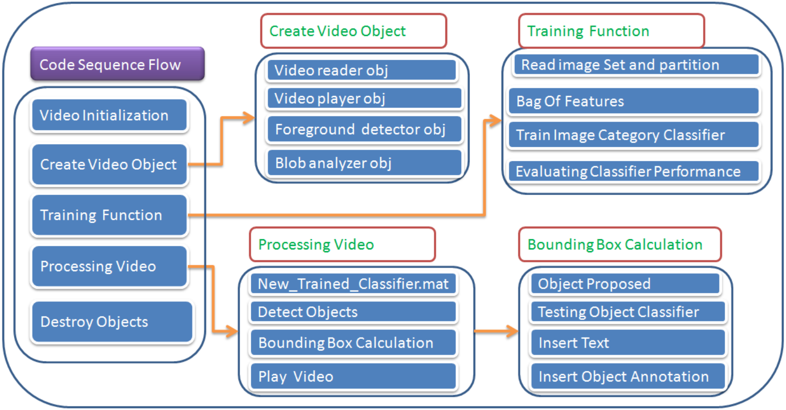
Approches:

- Object proposed
- Test Object Classifier
- Bounding Box

Approches:

- Object proposed
- Test Object Classifier
- Bounding Box

Code Sequence Flow:



Code Sequence Flow:

```
'object is'   'Motorbikes'   ',object id '   '2'   ',present frame'   '133'   ',no_of_obj in frame'   '2'
'object is'   'Leopards'   ',object id '   '1'   ',present frame'   '134'   ',no_of_obj in frame'   '2'
'object is'   'lion'   ',object id '   '2'   ',present frame'   '134'   ',no_of_obj in frame'   '2'
'object is'   'ferry'   ',object id '   '1'   ',present frame'   '135'   ',no_of_obj in frame'   '2'
'object is'   'car'   ',object id '   '2'   ',present frame'   '135'   ',no_of_obj in frame'   '2'
'object is'   'ferry'   ',object id '   '1'   ',present frame'   '136'   ',no_of_obj in frame'   '2'
'object is'   'car'   ',object id '   '2'   ',present frame'   '136'   ',no_of_obj in frame'   '2'
'object is'   'car_side'   ',object id '   '1'   ',present frame'   '137'   ',no_of_obj in frame'   '2'
'object is'   'car'   ',object id '   '2'   ',present frame'   '137'   ',no_of_obj in frame'   '2'
'object is'   'ferry'   ',object id '   '1'   ',present frame'   '138'   ',no_of_obj in frame'   '2'
'object is'   'car'   ',object id '   '2'   ',present frame'   '138'   ',no_of_obj in frame'   '2'
'object is'   'car'   ',object id '   '1'   ',present frame'   '139'   ',no_of_obj in frame'   '2'
'object is'   'car'   ',object id '   '2'   ',present frame'   '139'   ',no_of_obj in frame'   '2'
'object is'   'ferry'   ',object id '   '1'   ',present frame'   '140'   ',no_of_obj in frame'   '2'
'object is'   'car'   ',object id '   '2'   ',present frame'   '140'   ',no_of_obj in frame'   '2'
'object is'   'car'   ',object id '   '1'   ',present frame'   '141'   ',no_of_obj in frame'   '2'
'object is'   'Motorbikes'   ',object id '   '2'   ',present frame'   '141'   ',no_of_obj in frame'   '2'
```

[1] D. G. Lowe, *"Object recognition from local scale-invariant features," International Conference on Computer Vision, Corfu, Greece, pp. 1150-1157,September,1999.*

[2] D. G. Lowe, *"Distinctive image features from scale-invariant keypoints," International journal of computer vision 60 (2), 91-110,2004.*

[3] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, *"Visual categorization with bags of keypoints," Workshop on statistical learning in computer vision, ECCV 1 (1-22), 1-2,2004*

[4] D. Nister and H. Stewenius, *"Scalable Recognition with a Vocabulary Tree," Center for Visualization and Virtual Environments,Department of Computer Science, University of Kentucky 2006*

[5]

Caltech101., *"'http://www.vision.caltech.edu/Image_Datasets/Caltech101/"*