PROJECT 3

MODEL SELECTION AND EVALUATION

PROJECT 3 DESCRIPTION

- Carry out model evaluation and selection for predictive analytics on image data.
- Evaluate different modeling/analysis strategies and decide what is the best.
- Present sound evidence in the form of model assessment, validation and comparison.
- Communicate your decision and supporting evidence clearly and convincingly in an accessible fashion.

NOT A COMPETITION OF PREDICTION ACCURACY

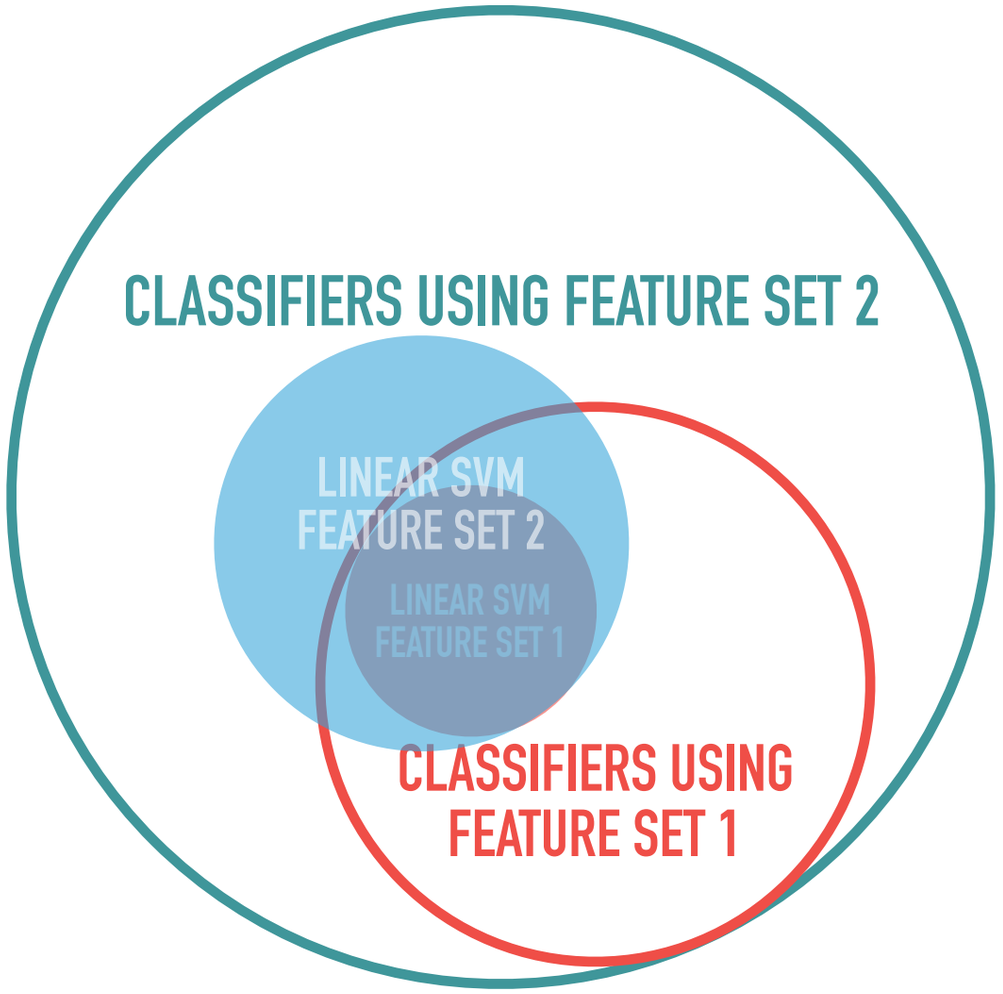
BASELINE MODEL

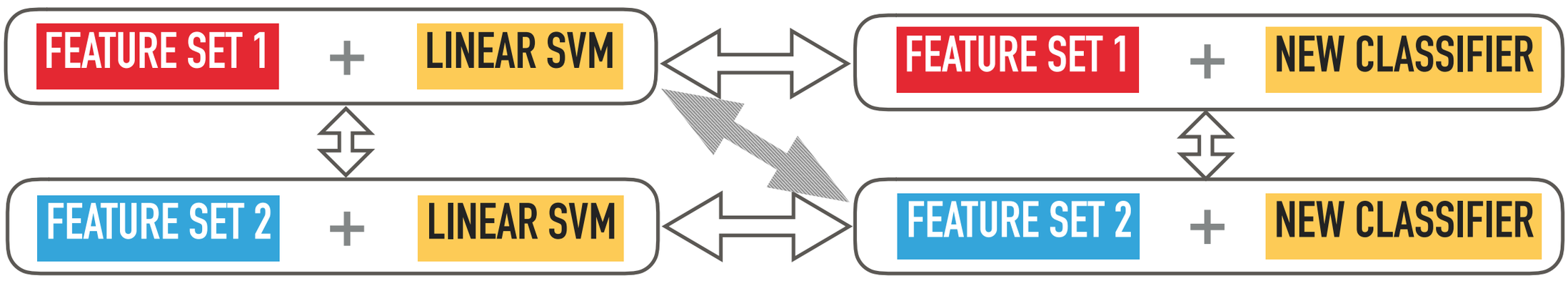
- Using color histogram (or color distribution) as features.
- Linear SVM as classifier.
- Task 1 will be implementing this strategy and tune it correctly.

PROPOSED STRATEGY

- Consider better features
- Consider better models
- Comparison should be structured to establish the values added by
  - New features
  - New method
- Nested model comparison structure.


FEATURE SET 2
FEATURE SET 1


CLASSIFIERS USING FEATURE SET 2
LINEAR SVM FEATURE SET 2
LINEAR SVM FEATURE SET 1
CLASSIFIERS USING FEATURE SET 1


FEATURE SET 1
+
LINEAR SVM
FEATURE SET 2
+
LINEAR SVM
FEATURE SET 1
+
NEW CLASSIFIER
FEATURE SET 2
+
NEW CLASSIFIER

SUBMISSION OF PROJECT 3

- Submission
  - A well-documented GitHub repo (following instruction given in the starter codes).
- a file of feature processing codes (**feature.R**) should take
  - an input folder containing images
  - outputs a folder of “feature” objects with features for each image
    - RData, or other R readable file
    - Each file should have the same name as the corresponding image.

SUBMISSION OF PROJECT 3

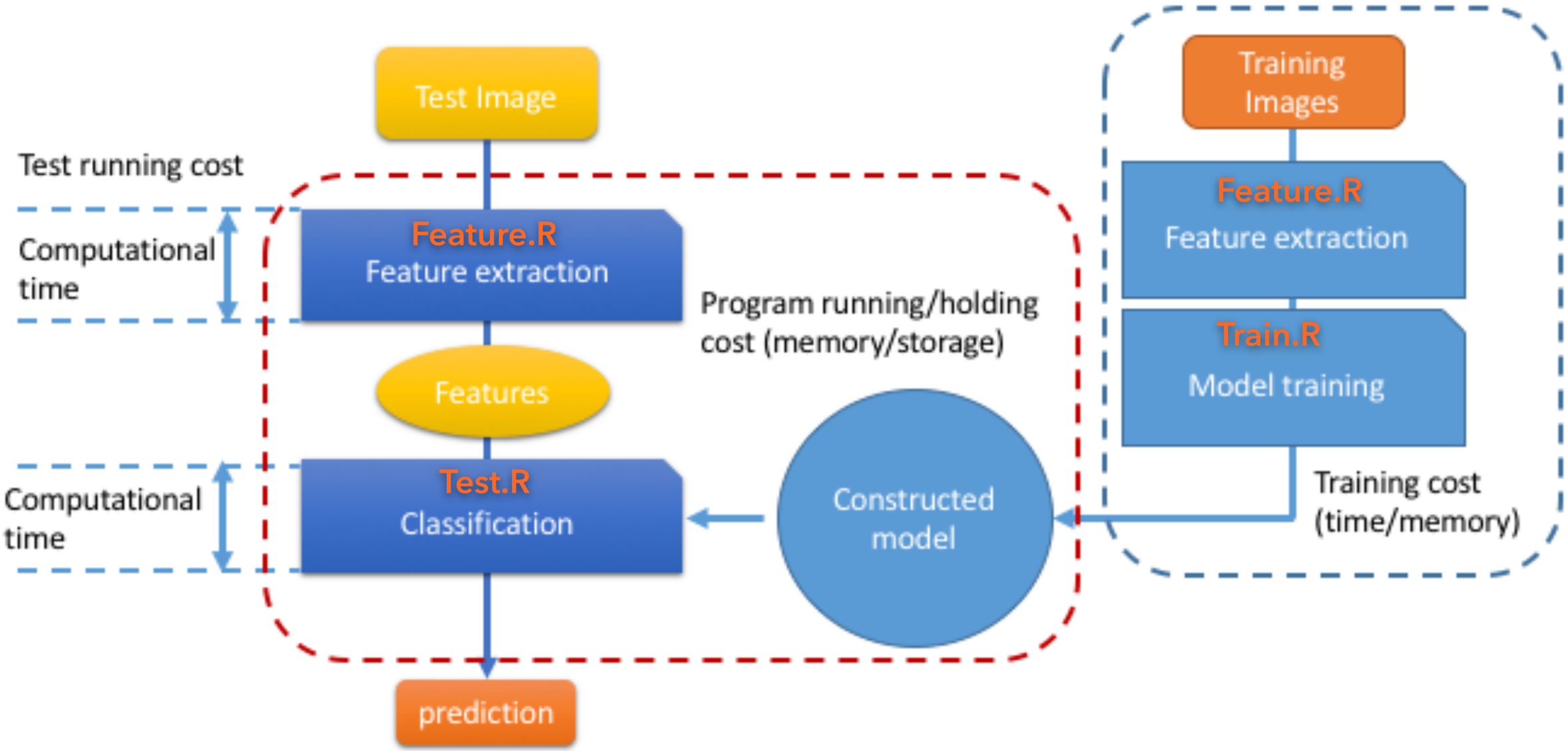
- Submission
  - A well-documented GitHub repo (following instruction given in the starter codes).
    - a file of training codes (**train.R**) should take
      - a folder of input feature objects (for training images)
      - a CSV file contains image names and labels
      - outputs two “model” objects (RData, or other R readable file); One for the baseline model and one for the new model.
      - model training should include tuning.

SUBMISSION OF PROJECT 3

- Submission
  - A well-documented GitHub repo (following instruction given in the starter codes).
  - a file of testing codes (**test.R**) should take
    - a folder of input feature objects (for testing images)
    - an input “model” object from train.R
    - output predicted labels

SUBMISSION OF PROJECT 3

- You can use any methods to generate features
- On March 23rd, we will first fork all project repos to save a time-stamped version of all your codes.
- On a new set of 2000 images (similar sizes as the data set used for project 3), each team will have 30 minutes to process them into features chosen.
- Submit the process features as a folder of feature objects file. The feature objects should be readable by train.R and test.R.
- We (the instruction team) will then run an **evaluation.R** file on all submissions.


Test Image
Test running cost
Computational time
Feature.R
Feature extraction
Program running/holding cost (memory/storage)
Features
Test.R
Classification
Computational time
Constructed model
prediction
Training Images
Feature.R
Feature extraction
Train.R
Model training
Training cost (time/memory)

SUBMISSION OF PROJECT 3

- You should also prepare **a presentation** for this project
  - Methodology details of the proposed solution
  - Evaluation results as supporting evidence
    - Prediction performance comparison between baseline and new models.
    - Time/cost analysis.

EVALUATION OF PROJECT 3

- Peer rating on presentations 20%
- By instructional team
  - Presentation 40%
    - Methodology
    - Interpretability of the features selected (scientific insights?)
    - Presentation
- Reproducibility evaluation based on the 2000 new images 40%
  - Comparing training error rate and test error rate (0-1 loss will be used) on the 2000 new images.
  - Comparing stability of the methodology comparison results on baseline and new models (using cross-validation methods on the 2000 new images)
  - Consistency between the results from new images with the results from the project images (in terms of percent of improvement)

TUTORIALS

- [Today] Image features
- [Next week] Cross-validations, avoid overfitting