Can You Unscramble a Blurry Image?

GR5243 section 1 Group 5
Sizhu Chen, Hengyang Lin, Yang Yue,
Bailey Pierson, Binglun Zhao

Content

Agenda

★ Use Gradient Boosting Machines (GBM) to enhance the resolution of images -- **Baseline**

★ Make improvement on baseline model

★ Apply other models to build a new method for Super Resolution

★ Choose the best model -- **Advanced Model**

★ Compare results of baseline model and advanced model

Model Selection

Baseline Model

Key points Feature

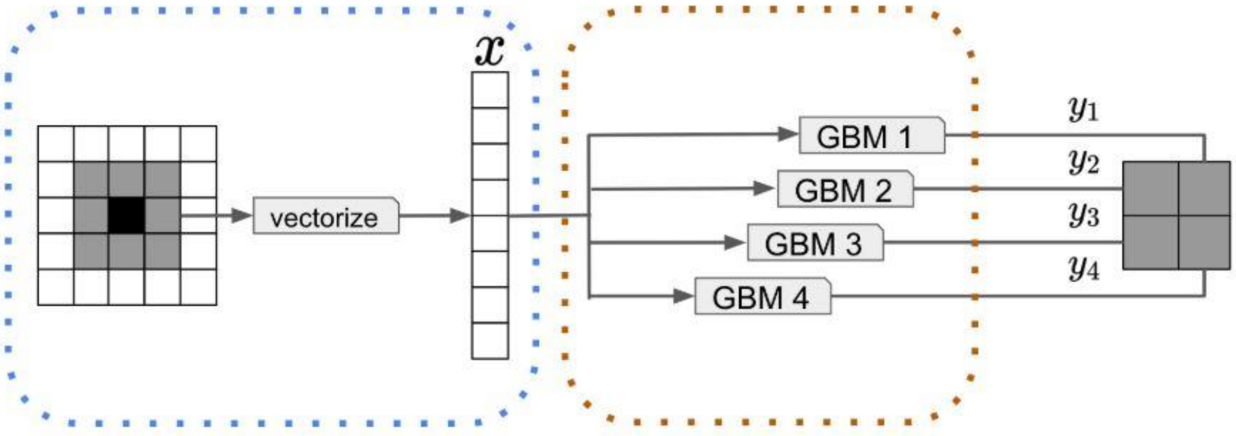
XGBoost

Super-Resolution CNN

Baseline Model (GBM)

<u>Feature Extraction:</u> Patch-based Method, random sample

Sample 1000 points from LR images randomly, then take each point's 3x3 patch as feature



Running time for 1500 pairs: 106.93 s

Baseline Model (GBM)

Parameter Tuning:

n.trees = 200

Interaction.depth = c (3, 5)

Shrinkage = 0.1

Bag.fraction = 0.5

	mean (cv.error)	std (cv.error)
depth = 3	0.001745954	1.314451e-05
depth = 5	0.001713095	3.714368e-06

Baseline Model (GBM)

Main

Output Picture (Original LR vs Predicted Picture):

Improvement Method I -- Keypoints Feature

Feature Extraction: Keypoints Detection

When extracting points from images, instead of randomly sampling, we give more sample weight to areas where color changes drastically, e.g. corners, boundaries.

Feat

Improvement Method II -- XGBoost

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance, much faster than regular Gradient Boosting.

XGB

Main

How to evaluate performance ?

We have 2 kinds of feature data as train data.

1. Random Selected Sample
2. Key points Sample

We have 2 kinds of fit methods.

1. GBM
2. XGBoost

Cross validation error (Test error) is used to evaluate different models based on same train data.

Because the pixels at key points changes more dramatically than plain points, it would be more difficult to fit the value at key points. So it’s unfair to compare cross validation error directly.

We use PSNR between predicted pictures and true HR pictures to evaluate each following situation.

1. Random Sample + GBM (Baseline)
2. Key points Sample + GBM
3. Random Sample + XGboost
4. Key points Sample + XGboost

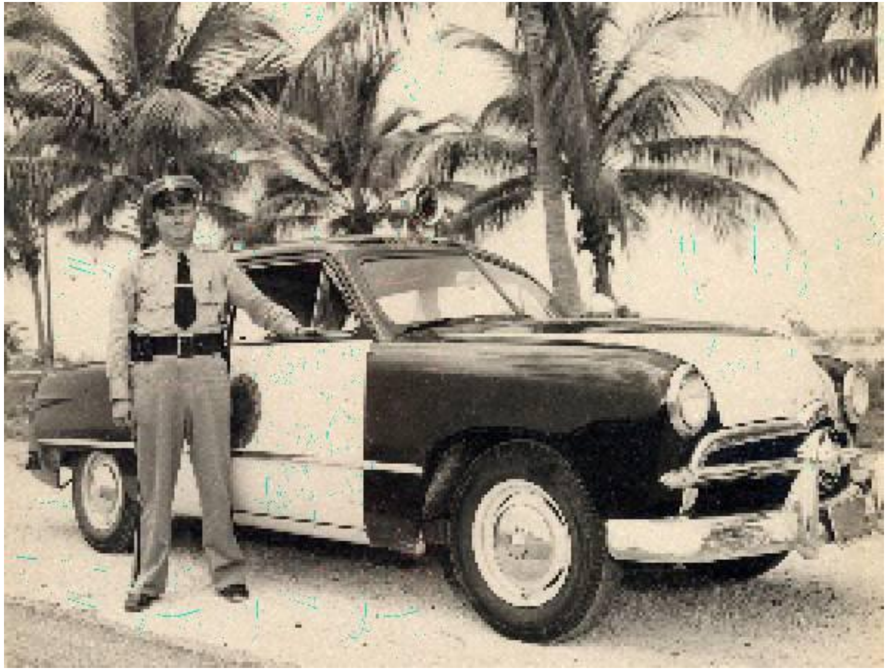
Improvement Method Results (10 pictures)

	PSNR	Running Time
Baseline Model	21.83	1338.9 + 5.79*10
Keypoints Feature + GBM	14.95	1342.5 + 5.29*10
Random Feature + XGBoost	23.26	601.83 + 9.53*10
Keypoints Feature + XGBoost	23.51	849.24 + 9.53*10

Main XGB

Random Selected Sample + GBM

Output Picture (Original HR vs Prediction):



Keypoints Sample + GBM

Output Picture (Original HR vs Prediction):

Random Selected Sample + XGBoost

Output Picture (Original HR vs Prediction):

Keypoints Sample + XGBoost

Output Picture (Original HR vs Prediction):

Improvement Method Results (10 pictures)

	PSNR	Running Time
Baseline Model	21.83	1338.9 + 5.79*10
Keypoints Feature + GBM	14.95	1342.5 + 5.29*10
Random Feature + XGBoost	23.26	601.83 + 9.53*10
Keypoints Feature + XGBoost	23.51	849.24 + 9.53*10

Further thoughts

If we are going to expand the volume of train data, there are two ways:

1. Expand the number of points on each image from 1000 to 3000.

2. Expand the number of features on each point, from 8 to 24 while number of points remains. (Surrounding 8 grids to surrounding 24 grids)

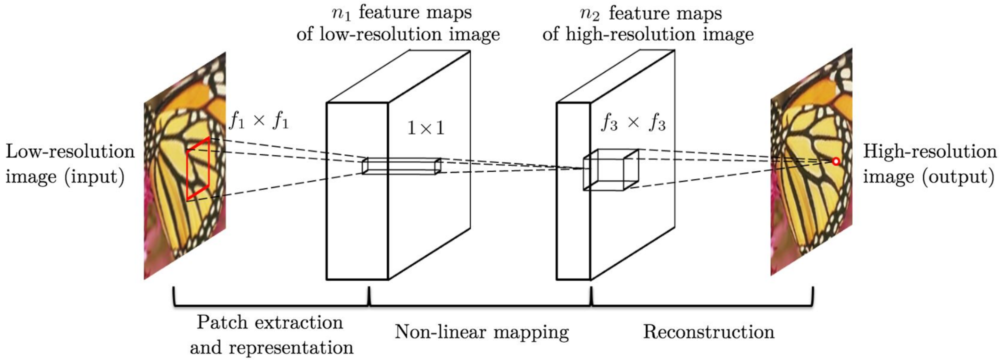
In both way, the volume of training data increase by 3 times.

But which one is better ?

If time permits, we would love to explore this question.

Advanced Method -- Super-Resolution CNN

1. Used bicubic interpolation(openCV) to resize LR image to desired size. (in our case 2 times larger)
2. Applied 3 convolutional layers with filter size 9x9 for the first one, 1x1 for the second layer, and 3x3 for the last one. (not very deep actually)
3. It's slow, and training with GPU is highly recommended which will only take around 1 hour. However, it has the best performance.



Advanced Method -- CNN

Output Picture (Origianl HR vs CNN Prediction):

Final Model Selection

Baseline Model:

n_points = 1000

n.trees = 200

interaction.depth = 5

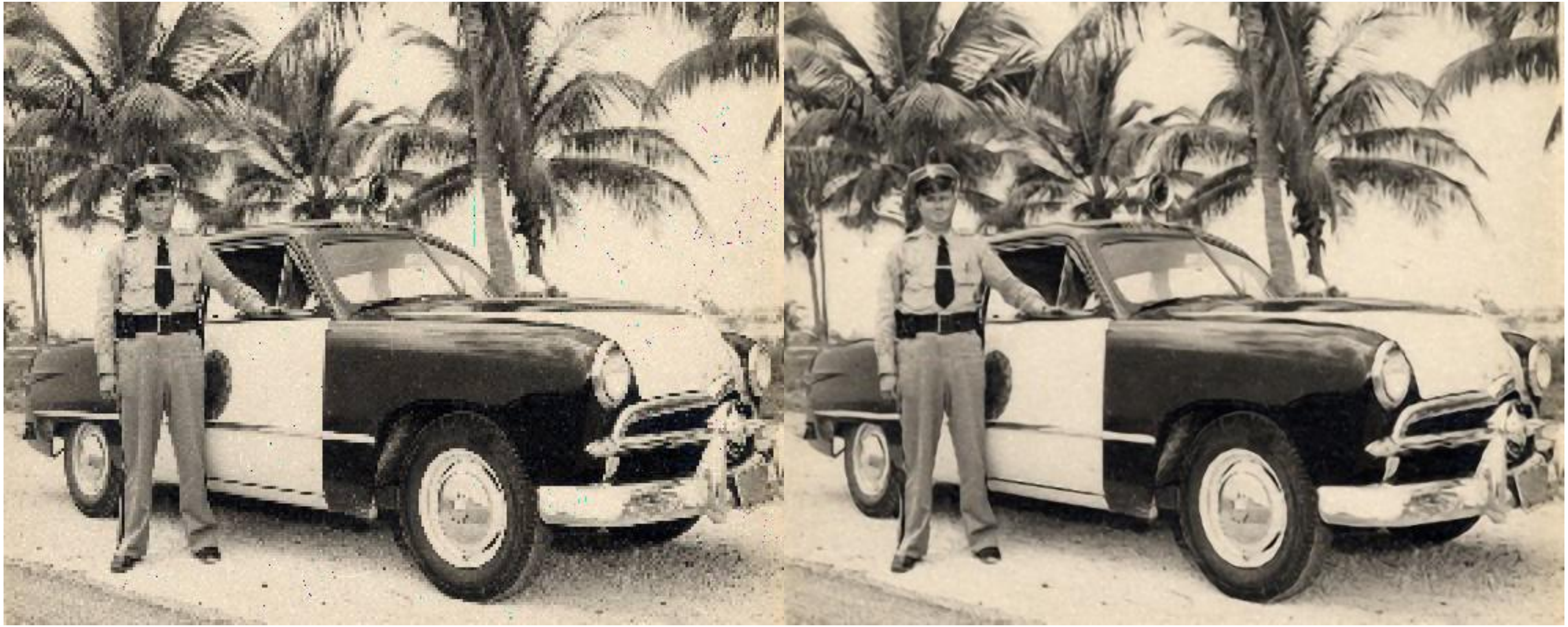
shrinkage = 0.05

bag.fraction = 0.5

Advanced Model:

Super-Resolution CNN

Result Compare (Baseline vs SRCNN)



Result Compare (Baseline vs SRCNN)

	PSNR	Running time
Baseline	21.83	5.8s/pic
SRCNN	24.62	3.0s/pic

Thank you ! :)