AdaFilter: Adaptive Filter Design with Local Image Basis Decomposition for Optimizing Image Recognition Preprocessing

Paper ID: #704
PS T5.5

Aiga Suzuki[*†‡], Keiichi Ito[*], Takahide Ibe[*], Nobuyuki Otsu[*‡]
[*]Adacotech Inc., [†]University of Tsukuba, [‡]National Institute of Advanced Science and Technology, Japan
contact: ai-suzuki@aist.go.jp

A novel optimal image preprocessing filter design method for any image recognition tasks with the image basis decomposition and the Black-box optimization

Background

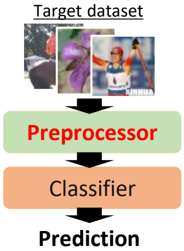
Image Preprocessing in Recognition Tasks

- **An appropriate image preprocessing** strongly affects the model performance in image recognition tasks in broad tasks
  - e.g., the **blurring** preprocessing to **reduce an image noise**
  - e.g., the **edge extraction** in the **shape-dominant** recognition tasks
- **Linear image filtering** is the most common way to preprocessing

Optimal Preprocessing Design

- **Designing an optimal way to preprocess** images are important to improve the system's accuracy
- Optimal preprocessing design has been achieved by **finding the combination of typical filters**

✗ Lack of representability in just a combination

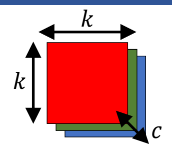


Our Research Question

How can we design an optimal preprocessing filter in pixel-level?

Objective and Main ideas

Pixel-level Design an Optimal filter



- Let R be an expected generalization performance, designing an optimal filter f is written as

$$\text{maximize } R(f)$$

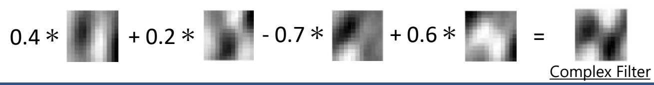
- **But this optimization problem is in very high-dimensional space**
  - e.g., the search space of k=15, RGB-colored filter is in the $\mathbb{R}^{675}$

[Our goal] To realize to design the pixel-level optimal filter which maximize the generalization performance for given task

Designing a Filter as a Linear Combination of Bases

- **Image filtering** is essentially **emphasizing/inhibiting** local patterns
  - Core idea of *Neocognition*, that is the *origin of modern CNNs*

[Main idea] To represent a preprocessing filter kernel as a Linear combination of Local image bases (a.k.a., Image atom)



Methodology: Black-box Filter Optimization

Image Basis Decomposition

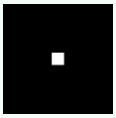
- First, $k \times k$ **local regions will be sampled** from the training dataset to calculate the image bases; Let this N'sampled matrix be $X' \in \mathbb{R}^{N' \times k^2}$
- ***The filter bases* are calculated as a decomposed components** with the low-rank approximation *(the same manner as a sparse modeling)*
  - This study uses **Sparse PCA** and **Independent Component Analysis (ICA)**

Injecting *"Unit" Filter* to the Filter Bases

- By *just* adopting decomposed components as a filter bases, **the designed filter sometimes works *"too hard"***
  - When the raw images were already sub-optimal, they *should be* ***ignored***
- The *"unit"* filter, a.k.a., *"2-dimensional discrete impulse"*, is added to the filter bases **to ensure the representability of *"Do nothing"***
  - The unit filter is defined mathematically as…

$$\mathbf{1}(x, y) = \begin{cases} 1 & (x = y = \lceil k/2 \rceil) \\ 0 & (\text{otherwise}) \end{cases}$$

that means like



Reformulation of the Optimization Problem

- Assume that the **optimal filter can be represented as a linear form of calculated dataset's local image bases,** a filter f is written as:

$f = V\boldsymbol{a}$; where $V \in \mathbb{R}^{k^2 \times M}$ is M bases, and $\boldsymbol{a} \in \mathbb{R}^M$ is a coefficients

- Then the **original k^2c-dimensional filter optimization** problem can be **relaxed into M-dimensional coefficients optimization**

$$\underset{f}{\text{maximize}}\ R(f) \Rightarrow \underset{\boldsymbol{a}}{\text{maximize}}\ R(V\boldsymbol{a})\ \ sbj.to\ \ a_i \in [-1, 1]$$

Solving the Coefficient Optimization

- How to maximize the expected generalization performance R?
  - There is **no information about R** ⟹ We must solve it as **Black-Box**
  - The expected generalization performance is approximated by **result on validation data** (hold-out / K-fold cross validation etc…)

Black-Box optimization algorithms

- **Random-search**
- **Grid-search**

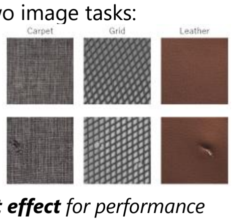
✓ **Bayesian optimization (e.g., TPE)**
✓ **Evolution Strategy (e.g., CMA-ES)**

Experiments

We applied the proposed preprocessing for two image tasks:

1. Anomaly detection with non-DL model

Method: HLAC feature + Subspace method
Dataset: MVTec Anomaly Detection dataset
Metrics: Validation ROC-AUC score



➢ *The task that* ***preprocessing have dominant effect*** *for performance*

2. Classification problem with CNN-feature

Method: Pretrained-ResNet50 + Linear-SVM
Dataset: Caltech-101 image dataset
Metrics: Validation F-measure

➢ *n.b., CNNs are* ***optimized for non-filtered images***

We compared generalization performance for test data between the typical preprocessing image filters and the proposed method

Results and Discussion

Results on MVTec anomaly detection (HLAC+SM) ($k = 15, M = 16$)

ROCAUC Task	Best score on typical	ICA+ES	sPCA+ES	ICA+TPE	sPCA+TPE
Carpet	*0.648*	**0.843**	0.717	0.816	0.722
Grid	*0.732*	0.908	**0.913**	0.891	0.901
Leather	*0.916*	0.965	0.954	**0.971**	0.969

Results on Caltech-101 classification (ResNet-50 + SVM) ($k = 15, M = 16$)

F1-score	No-filter	ICA+ES	sPCA+ES	ICA+TPE	sPCA+TPE
Carpet	0.865	**0.875**	*0.839*	**0.882**	*0.832*

✓ **The proposed method showed significant improvements** not only for HLAC+SM but **also for CNN**, that is very sensitive case

▷ We aim to **apply the method to non-image tasks, e.g., 1D-signals**

The proposed method can be a leading preprocessing design way because of its performance and model-data-agnostic property