Learning Feature Descriptors Using CapsNets and Deep Convolutional Neural Networks

Ashish Rao, Mai Lan Nguyen, Melinda Zhu | Mentor: Yinan Zhang

CS231N
Spring 2022

Background

Problem: Detecting keypoints and tracking them across images is a fundamental task in Computer Vision. Used in computer vision tasks such as visual odometry and SLAM

Challenges: difficult due to images of the same item being obstructed, taken from any angle, with any lighting and scale (need scale and rotation invariance)

Previous attempts: hand-crafted SIFT, state-of-the-art Triplet Loss and Siamese Network

New Approach: CNNs - ResNet (baseline) and Capsule Network

- Experimentation with network depth for keypoint description
- Utilizing the spatial structure preservation of Capsule Networks

Problem Statement

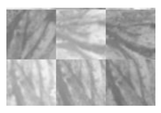
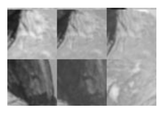
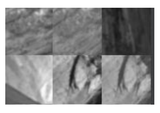
Input: 32x32 image patch centered around a keypoint from MVS dataset

Output: Vector representing a feature descriptor for that keypoint

Approach: Using Capsule Networks/CNNs to produce feature descriptor vector for a given patch

Data

32x32 patches from Statue of Liberty, Yosemite, & Notre Dame, with labels for which patches are centered around same keypoint: 500k patch pairs extracted around keypoints; 100k pairs used for eval.

References

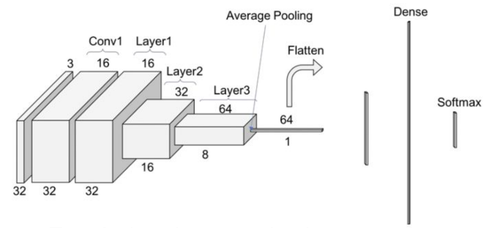
[1] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.
[2] Geoffrey E. Hinton, Nicholas Frosst, and Sara Sabour. Dynamic routing between capsules. 31st Conference on Neural Information Processing Systems, 2017.
[3] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. *Proceedings of the British Machine Vision Conference* (BMVC), pages 119.1–119.11, 2016

Methods

Method 1: SIFT [1]

- Locate keypoints in images, and craft feature descriptors by computing gradient magnitude and orientation around keypoint
- Hand-crafted instead of learning-based feature descriptors

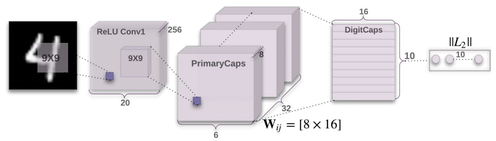
Baseline: ResNet



- Transfer learning: pretrained on ImageNet/CIFAR-10, finetuned on MVS grayscale images. Modified network to adapt to 32x32 inputs.

Method 2: CapsNets [2]

- Shallow CNN where lower-level capsules encode detailed features (e.g. eyes, mouth) and higher-level capsules encode larger classifications (e.g. face).
- **Dynamic routing** instead of max pooling: use the probability that a feature encoded in a capsule is present in the input.



Method 3: Tfeat CNN [3]

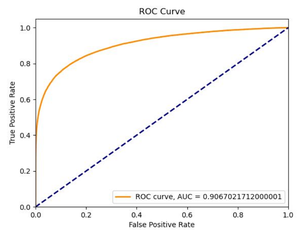
- State-of-the-art model using **triplet loss**: training with the goal of getting two patches of the same keypoint as close as possible in the feature space, and getting the third patch of a different keypoint farther away from the pair.

Results

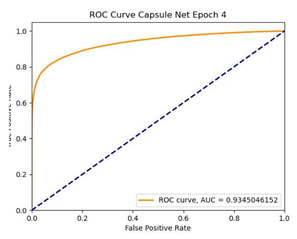
(right) ROC curves for SIFT, ResNet, Capsule Network, and Tfeat feature descriptors. The tables below summarizes the AUC and FPR95 achieved by these methods, as well as by different variants of the ResNet model.

Method	AUC	FPR95
SIFT	0.9067	0.3475
ResNet-18	0.7527	0.3731
Capsule Net	0.9345	0.3134
TFeat	0.9818	0.0873

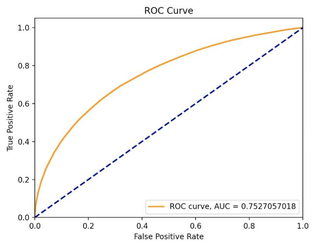
Model	AUC	FPR95
ResNet-18	0.7527	0.3731
ResNet-34	0.7562	0.3886
ResNet-50	0.7512	0.4532
ResNet-101	0.7462	0.4532



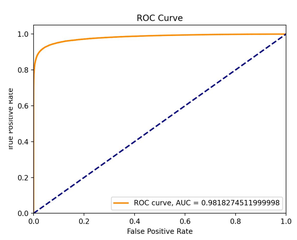
SIFT ROC curve



Capsule Network ROC Curve



ResNet ROC Curve

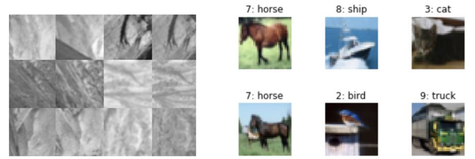


TFeat ROC Curve

Analysis

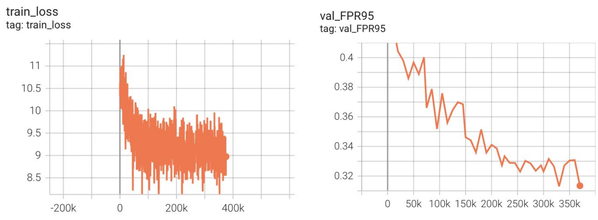
The CapsNet outperforms the SIFT and ResNet baselines, but fails to outperform the TFeat model. The CapsNet may be outperforming ResNet and SIFT due to the dynamic routing algorithm used and its ability to preserve spatial relationships between detected features.

ResNet performs the worst, and adding more layers degrades performance, perhaps because deep networks are better for detecting high level features, but shallow networks are better for detecting low level features relevant to producing feature descriptors. The pretraining dataset used for the ResNet models (CIFAR 10) is also very different from the Phototour dataset.



Comparison of datasets (left: phototour, right: CIFAR-10)

Indications of overfitting in the CapsNet are not present; both train and validation metrics steadily improve and eventually plateau.



Train loss and validation FPR95 for CapsNet while training

Conclusions & Future Work

Capsule Networks outperform popular hand-crafted **SIFT** method and baseline **ResNet** method for keypoint feature description across both AUC and FPR95 metrics

CapsNet does not outperform the current state-of-the-art learning based method (TFeat model), but is still a **competitive choice**.

Shallower networks perform better at the keypoint description.

Future work: performing hyperparameter tuning, performing Neural Architecture Search to optimize the model, or investigating modifications to the Capsule Network architecture such as an alternative to the agreement-based routing algorithm used here.

We would like to acknowledge our mentor Yinan Zhang and course staff for support and guidance + AWS for computing resources.