

# 해싱을 이용한 메모리 최적화 NBNN 이미지 분류기

이윤석<sup>o</sup>, 윤성의

카이스트 전산학부

ys.lee@kaist.ac.kr, sungeui@gmail.com

## 요약

NBNN은 이론적으로 간결하며 구현이 용이한 최근접 이웃 기반의 이미지 분류기로 표현자 양자화를 거치지 않아 표현자의 분별력을 잘 유지하며 클래스의 특성을 일반화하는 능력이 뛰어나지만, 데이터 양이 많아지면 질의 시간이 오래 걸리는 문제를 안고 있다. 본 논문에서는 NBNN과 그 후속 연구에 Spherical hashing을 적용하고, 이진 코드의 최근접 이웃 연산에 적합한 계층적 인덱싱 방법을 제안하여 확장성 문제를 해결하고자 한다. 제안된 해싱 기법을 적용하면 이전의 연구와 비교하여 분류 정확도와 질의 시간은 비슷한 수준으로 유지하면서 메모리 사용량을 수십 배 가량 감소시킬 수 있다. 또한, 확장성 문제가 더욱 심해지는 고차원 표현자를 이용한 이미지 분류 작업에서 이러한 이점이 더욱 극대화될 수 있다.

## 1. 서론

이미지 분류 방법은 컴퓨터 비전 분야에서 오랫동안 중요한 연구 주제로 취급되어왔으며, 현재까지 활발히 연구가 진행되고 있다. 이미지 분류는 질의 이미지(query image)가 주어지면, 해당 이미지가 어느 클래스에 속하는지를 판단하는 기술이다. 최근 deep CNN (Convolutional Neural Networks)[1]이 높은 정확도와 빠른 질의 속도에 힘입어 큰 관심을 받았으며, 후속 연구[2, 3]가 많이 진행되고 있다. 이렇게 최근 제안된 많은 이미지 분류 방법들이 CNN을 이용하지만, 그 이전에 연구되었던 분류기들을 완전히 대체할 수 있는 것은 아니다. 이전에 연구되었던 분류기를 CNN에 결합하는 방향[4]으로 연구가 진행되기도 하고, CNN의 오랜 학습 시간 때문에 이를 사용하는 것이 적합하지 않은 상황에서는 이전의 분류기를 단독으로 사용할 수도 있다.

CNN이 제안되기 이전의 이미지 분류기 중 대표적인 것으로 NBNN(Naïve Bayes Nearest Neighbor) 이미지 분류기[5]가 있다. NBNN은 CNN보다 일반적으로 정확도가 떨어지는 경향을 보이며 질의 속도에 있어서도 큰 이점을 보이지 않지만, 학습 과정이 필요 없어서 데이터만 준비되면 즉시 분류를 수행할 수 있다. 또한 이론적으로 간결하고 구현이 용이하기 때문에 필요한 상황에 맞추어 알고리즘 일부를 수정, 개선하여 사용할 수도 있다. [11] 하지만 최근 접 이웃(nearest neighbor) 기반 방법의 특성상 데이터

의 양이 많아지면 모든 데이터를 메모리에 불러오기 힘들어지고 질의 속도도 느려지는 문제가 있다.

본 논문에서는 해싱 기법을 이용하여 NBNN 이미지 분류기의 메모리 사용량을 줄이고 질의 속도를 향상시키는 방법을 제안한다. 본 논문의 구성은 다음과 같다. 2 장에서는 본 연구와 관련된 연구에 대해서 소개하고, 3 장에서는 해싱을 NBNN에 적용하는 방법에 대하여 소개한다. 4 장에서는 기존의 방법과 제안된 방법을 실험을 통해 비교하고 5 장에서 결론을 맺는다.

## 2. NBNN 이미지 분류기 및 관련 연구

기존의 이미지 분류기들이 대개 단어주머니(bag-of-words)와 같은 표현자 양자화(descriptor quantization) 기법을 이용하는 것과 달리 NBNN[5]은 코드북에 의존하지 않고 원본 이미지 표현자를 그대로 사용하여 표현자의 분별력을 유지한다. 또한, 일반적으로 이미지 대 이미지 거리를 측정하여 분류를 수행하던 이전의 방식과는 다르게 이미지 대 클래스(image-to-class) 거리를 측정하여 클래스의 특성을 일반화하는 능력을 갖는다. 이를 통하여 가지고 있는 데이터셋에 질의 이미지와 비슷한 이미지가 없더라도, 특정 클래스에 속한 이미지 전체에서 비슷한 지역 표현자를 탐색하여 성공적으로 분류를 수행할 수 있게 한다.

하지만 NBNN은 원본 이미지 표현자를 그대로

사용하고, 각각의 클래스에 속한 모든 표현자와 거리를 비교해야하는 점 때문에 대용량 데이터를 처리하기에는 문제가 있다. 이러한 문제를 해결하기 위해 제안된 Local NBNN[7]은 각각의 질의 이미지 표현자에 대해 모든 클래스와의 거리를 측정하는 대신 가까운 몇몇 클래스와의 거리만 측정하는 방법으로 확장성(scalability) 문제를 개선하였다.

이미지 분류의 확장성 문제를 해결하기 위한 다른 접근법으로 해싱 기법이 연구되어 왔다. 그 중 이진 코드화 해싱 기법은 고차원의 이미지 표현자를 짧은 길이의 이진 코드로 표현함으로써 메모리 사용량을 줄이고, 연산장치에서 매우 빠르게 수행할 수 있는 XOR 비트 연산만을 이용하여 거리 계산을 할 수 있다는 이점이 있다. 이러한 이진 코드화 해싱에 대표적인 연구로 Spherical hashing[6]이 있는데, 이 기법은 기존의 초평면 기반의 해싱 기법들과는 달리 초구 기반의 해싱 방법이기 때문에 높은 차원의 공간상에 닫힌 공간을 많이 만들어내어 이진 코

드화 이후에도 표현자간 유사도를 높게 보존할 수 있다.

### 3. 해싱을 이용한 NBNN

#### 3.1 NBNN 에 이진 코드화 해싱 적용

NBNN 이 어떻게 동작하는지 소개하기 전에 어떤 표기를 사용할지 정의하도록 한다. 임의의 표현자와 이미지, 클래스를 각각  $d, I, c$  라 하면 이미지  $I$  는 표현자들의 집합( $I = \{d_1, d_2, \dots, d_n\}$ )으로 나타낼 수 있다.

NBNN 에서 질의 이미지가 어떤 클래스에 속하는지 판단하기 위해서는 이미지 대 클래스 거리 ( $D_{I2c}$ )를 정의하여 측정해야 하는데, 이를 위해서는 먼저 표현자와 클래스( $D_{D2c}$ ) 사이의 거리를 정의해야 한다. 클래스  $c$ 에 속한 표현자 중  $d$ 와 거리가 가장 가까운 최근접 이웃 표현자를  $NN_c(d)$ 라 하면 표현자와 클래스 사이의 거리 및 이미지와 클래스 사

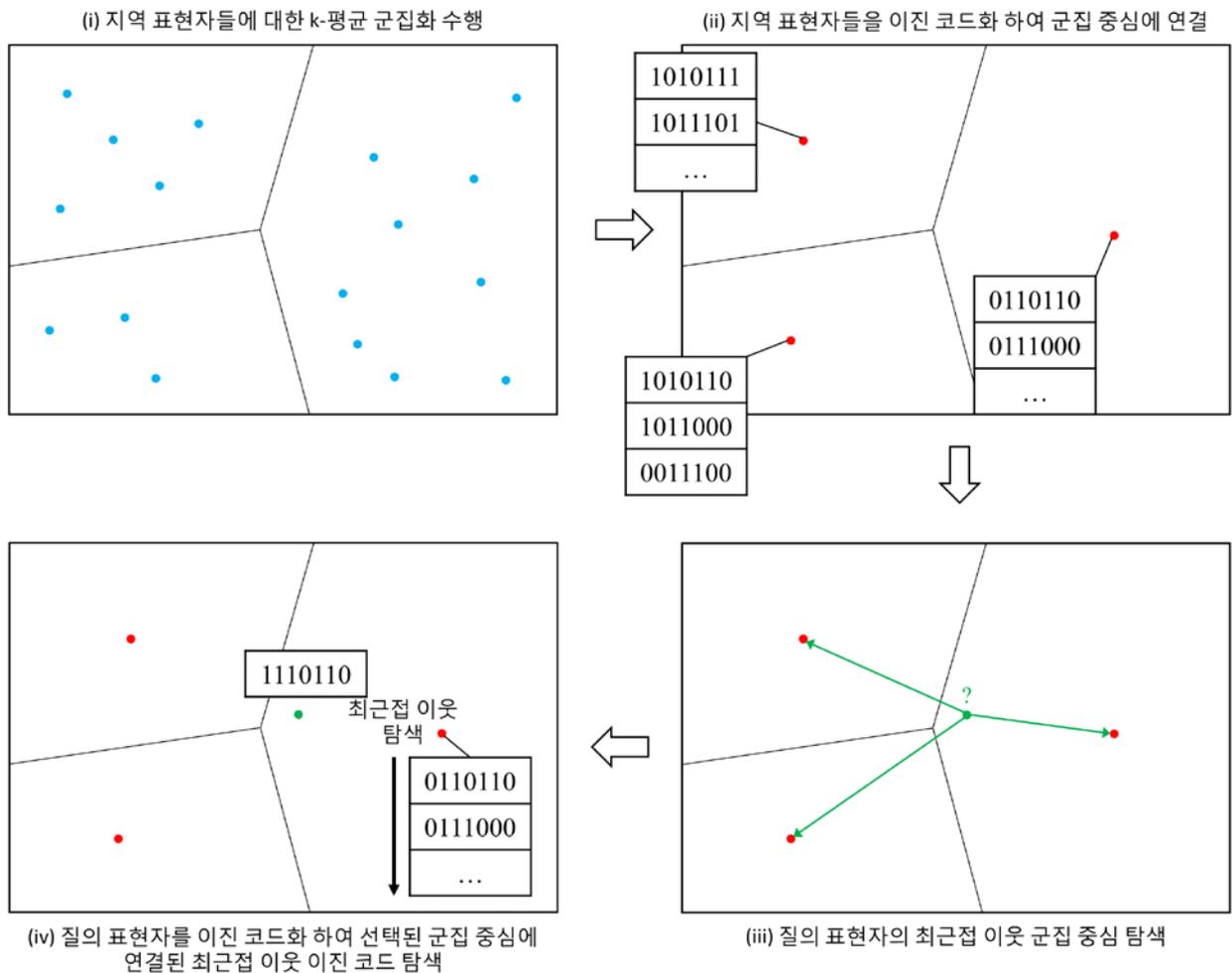


그림 1. 이진 코드화된 표현자의 최근접 이웃 연산을 위한 인덱싱 구축 방법. 파란색, 빨간색, 초록색 점들은 각각 학습용 이미지 표현자, 군집 중심, 질의 이미지 표현자를 의미한다.

이의 거리는 다음과 같이 정의된다.

$$D_{D2C}(d, c) = \|d - NN_C(d)\| \quad (1)$$

$$D_{I2C}(I, c) = \sum_{i=1}^n D_{D2C}(d_i, c) \quad (2)$$

이를 이용하여 임의의 이미지가 어떤 클래스에 속하는지는 아래의 식과 같이 판단할 수 있으며, 이는 [5]에서 Naïve Bayes 확률 모델에 근거한 최대 공산 분류기(maximum likelihood classifier)를 단순화하는 방법을 따른 것이다.

$$\hat{c} = \underset{c}{\operatorname{argmin}} D_{I2C}(I, c) \quad (3)$$

여기에 Spherical hashing 기법( $b = h(d)$ )을 적용하여 이미지 표현자를 이진 코드( $b$ )로 표현하도록 하였다. 이렇게 하면 이미지  $I$ 는 각각의 표현자들에 해싱 기법을 적용하여 얻어진 이진 코드들의 집합 ( $I_b = \{b_1 = h(d_1), b_2, \dots, b_n\}$ )으로 표현되며, 식(1)과 거의 유사한 방법으로 이진 코드와 클래스 사이의 거리( $D_{B2C}$ )를 정의할 수 있다.

$$D_{B2C}(b, c) = \operatorname{HammingDistance}(b, NN_C(b)) \quad (4)$$

또, 이를 이용하여 이미지와 클래스 사이의 거리 및 분류기를 유사한 방식으로 수정하여 표현할 수 있다.

### 3.2 이진 코드 데이터셋에 대한 인덱싱 방법

3.1 장에서 살펴본 것과 같이 NBNN에 해싱 기법을 적용해도 분류기를 나타내는 수식은 거의 동일하게 표현된다. 하지만 기존의 NBNN에서도 표현자에 대한 최근접 이웃 연산이 질의 시간의 대부분을 차지하며 확장성 문제를 야기하는 것처럼, 이진 코드에 대한 최근접 이웃 연산도 선형 탐색 방법으로 수행할 경우 과도한 횟수의 거리 계산으로 인해 비슷한 문제를 야기할 수 있다. 따라서 적절한 인덱싱을 통하여 최근접 이웃 연산을 빠르고 정확하게 수행해야 하는데, kd-트리에 기반을 둔 FLANN[10]과 같이 지역 표현자에 대한 인덱싱 방법은 이진 코드에 그대로 적용할 수 없다는 점이 문제가 된다. 그러한 인덱싱 방법은 표현자가 고차원의 벡터 형태로 이루어져 있다고 가정하는데, 이진 코드의 각 비트 값을 하나의 차원으로 취급하면 이진 코드화의 이점을 대부분 잃게 되기 때문이다.

이진 코드의 인덱싱 및 최근접 이웃 연산은 그림 1에 표현된 것과 같은 방법으로 수행한다. 우선 인덱싱 구조를 구축하기 위하여 (i) 가지고 있는 지

역 표현자들에 대한 k-평균 군집화(k-means clustering)를 수행한다. (ii) 표현자들에 해싱 기법을 적용하여 이진 코드화한 후, 각각의 군집 중심에 연결한다. 이 때, 이진 코드를 표현자들과 같은 공간상에 둘 수는 없기 때문에, 리스트 형태로 따로 저장한다. 인덱싱 구조 구축이 끝난 상태에서 질의 표현자가 들어오면, (iii) 질의 표현자와 가장 가까운 군집 중심을 탐색한다. (iv) 질의 표현자를 이진 코드화 하고, (iii)에서 선택된 군집 중심에 연결된 이진 코드의 리스트를 탐색하여 최근접 이웃을 찾는다.

위와 같은 계층적 방법을 사용하면, 이진 코드의 최근접 이웃을 찾기 위하여 최근접 이웃 연산을 두 번 수행해야 하지만, 각각의 연산의 대상이 되는 데이터의 숫자가 적어지고, 이진 코드간의 거리 계산은 표현자간의 거리 계산보다 훨씬 빠르기 때문에 이를 상쇄할 수 있다.

## 4. 실험 및 결과

본 연구에서 제안한 해싱이 적용된 NBNN의 성능을 확인하기 위하여 기존의 NBNN 및 Local NBNN과 비교 실험을 수행하였다.

### 4.1 실험 환경

이미지 데이터셋은 Caltech-101 [8]을 사용하였으며, background 클래스를 제외한 101 가지 클래스의 이미지를 사용하였다. SIFT[9]이미지 표현자를 지역 표현자로 사용하였는데, 일반적인 특징점 추출과정을 거치는 것이 아니라 이미지를 격자로 나누어 추출하였고, 다규모 구조(multi-scale)를 사용하였다. 이러한 환경은 이전 연구[5]를 참고하여 설정한 것이다. 각 클래스별로 15 개의 학습용 이미지와 15 개의 테스트용 이미지를 임의로 선택하여 사용하였다. Spherical hashing 이 적용된 경우, 이진 코드의 길이는 동일하게 64bit로 설정하였다.

이전 연구들은 실험을 위하여 논문을 참고하여 구현하였으며, 인덱싱 구조 구축을 위하여 FLANN[10]을 이용하였다.

### 4.2 실험 결과

NBNN에 Spherical hashing을 적용한 이미지 분류기(NBNN+SH)와 Local NBNN에 Spherical hashing을 적용한 이미지 분류기(Local NBNN+SH)를 각각 기존 방법인 NBNN[5], Local NBNN[7]과 비교하기 위하여 실험을 진행하였다. 그림 2는 이미지 분류 방법에 따른 정확도와 질의 시간을 측정하여 나타낸 것이

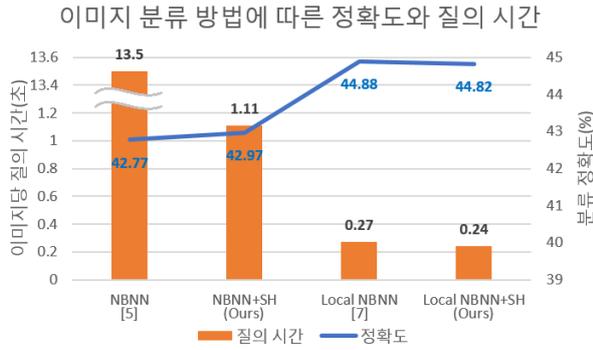


그림 2. 이미지 분류 방법에 따른 정확도와 질의 시간

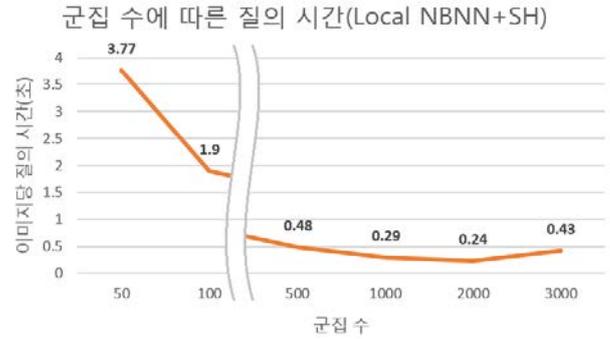


그림 4. Local NBNN+SH 에서 군집 수에 따른 질의 시간

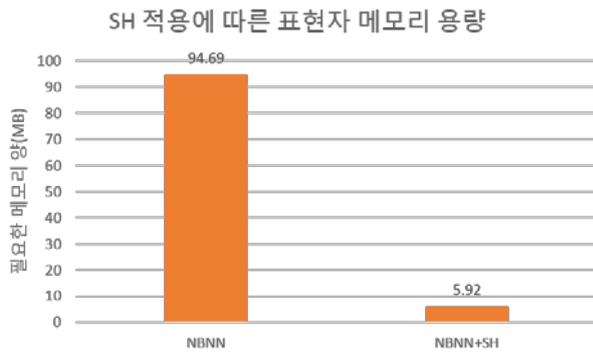


그림 3. SH 적용에 따른 표현자 메모리 용량. 표현자를 나타내는데 필요한 메모리 용량의 측면에서 Local NBNN 과 NBNN 간의 차이는 없다.

다. NBNN+SH 는 군집의 수를 30 개, Local NBNN+SH 는 군집의 수를 2000 개로 설정하였다.

NBNN 과 NBNN+SH 를 비교해 보면, 정확도는 그대로 유지하면서 속도가 10 배 이상 비약적으로 빨라진 것을 알 수 있다. 이는 표현자간의 거리를 계산함에 있어서, SIFT 표현자간의 거리 계산보다 이진 코드간의 거리 계산이 훨씬 빠른 이점의 영향을 많이 받은 것이다.

Local NBNN 을 Local NBNN+SH 와 비교해 보면, 정확도는 비슷하게 유지하였지만 속도도 그다지 큰 차이를 보이지 않는다. 한 쌍의 표현자간의 거리 계산은 NBNN 의 경우와 마찬가지로 훨씬 빠르지만, 사용한 이진 코드 인덱싱 구조의 특성상 훨씬 더 많은 힛수의 거리 계산이 필요했기 때문이다. 다만, 메모리 사용량에서 Spherical hashing 을 적용했을 때 큰 이점을 보이는데, SIFT 표현자 하나를 나타내는데 128byte 가 필요한 반면, 이진 코드는 8byte 의 메모리 만으로 표현할 수 있다. 인덱싱 구조 구축에 필요한 메모리를 제외하면, 이는 16 배 적은 메모리 만으로 이미지를 표현하면서도 분류 정확도와 질의 시간을 비슷하게 유지할 수 있는 것이다. 그림 3 은 주어진 실험 환경에서 표현자를 나타내기 위해 필

요한 메모리 용량을 보인 것이다. SIFT 가 비교적 낮은 차원의 이미지 표현자임을 감안하면, 고차원의 표현자에 대해서는 훨씬 더 큰 이점을 가질 것이다.

제한한 이진 코드의 인덱싱 구조의 효과를 확인하기 위하여 Local NBNN+SH 이미지 분류기를 사용하여 군집 수에 따른 질의 시간을 측정하였다. 모든 조건에 대한 분류 정확도는 43~46% 이내로, 의미 있는 차이를 보이지 않았다.

그림 4 에서 보이는 것처럼 군집의 수가 늘어날수록 이미지당 질의 시간은 줄어드는 경향을 보여 주었다. 하지만 군집의 수가 과도하게 많아질 경우 질의 시간은 오히려 증가하는 것을 볼 수 있는데, 이는 이진 코드간 거리 계산 횟수는 줄어들지만, 원하는 이진 코드가 속한 군집을 찾기 위한 군집 중심 탐색 시간이 증가하기 때문이다. 군집의 수가 적을 때는 군집 탐색 시간의 증가가 표현자간 거리 힛수 감소 효과와 상쇄되어 큰 영향을 끼치지 않지만, 군집의 수가 충분히 많은 경우 이미 표현자간 거리 계산에 걸리는 시간이 충분히 줄어들었기 때문에 군집 탐색 시간의 비중이 늘어나는 것이다.

### 5. 결론

본 연구에서는 NBNN 이미지 분류기에 Spherical hashing 을 적용하고, 이진 코드의 최근접 이웃 연산에 필요한 인덱싱 구조를 제한하였다. 제한한 방법은 NBNN 에 기반을 둔 최신의 이미지 분류기와 유사한 분류 정확도와 질의 시간을 보여주었으며, SIFT 이미지 표현자를 그대로 사용하는 것과 비교하여 16 배 정도의 메모리 사용량 감소 효과를 보여주었다. 이러한 이점은 이미지 표현자가 더 높은 차원을 갖게 될수록 더 뚜렷하게 나타날 것이며, NBNN 계열의 이미지 분류기의 확장성 문제를 해결하는데 큰 도움이 될 것이다.

## 감사의 글

본 연구는 미래창조과학부 및 정보통신기술연구진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [R0126-15-1108, 대용량 이미지 검색과 시제품 렌더링을 위한 근접질의 SW 개발]

논문 검토를 도와주신 윤웅직, 김동혁, 김수민, 권용선, 조재형님께 감사 드립니다.

## 참고문헌

- [1] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Advances in Neural Information Processing Systems (NIPS), pp. 1097-1105, 2012.
- [2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580-587, 2014.
- [3] N. Zhang, J. Donahue, R. Girshick, T. Darrell, "Part-Based R-CNNs for Fine-Grained Category Detection", European Conference on Computer Vision(ECCV), pp.834-849, 2014.
- [4] L. Xie, R. Hong, B. Zhang and Q. Tian, "Image Classification and Retrieval are ONE", International Conference on Multimedia Retrieval (ICMR), pp. 3-10, 2015.
- [5] O. Boiman, E. Schechtman, and M. Irani, "In defense of Nearest-Neighbor based image classification", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-8, 2008.
- [6] J.-P. Heo, Y. Lee, J. He, S.-F. Chang, and S.-E. Yoon, "Spherical hashing", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2957-2964, 2012.
- [7] S. McCann, and D. G. Lowe, "Local Naïve Bayes Nearest Neighbor for Image Classification", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3650-3656, 2012.
- [8] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories", IEEE. CVPR , Workshop on Generative-Model Based Vision, pp.178, 2004.
- [9] D. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, pp. 99-110, 2004.
- [10] M. Muja, and D. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration", In VISAPP International Convergence on Computer Vision Theory and Applications, pp. 331-340, 2009.
- [11] T. Tommasi, and B. Caputo, "Frustratingly Easy NBNN Domain Adaptation", IEEE Conference on Computer Vision (ICCV), pp.897-904, 2013.