

# 딥러닝 학습용 집적화된 데이터 증강 자동화 도구 개발

\*장찬호 \*이서영 \*박구만

서울과학기술대학교 전자IT미디어공학과

\*jchanho29@seoultech.ac.kr

## Development of integrated data augmentation automation tools for deep learning

\*Jang, Chan-Ho \*Lee, Seo-Young \*Park, Goo-Man

Seoul National University of Science And Technology

### 요약

4차 산업혁명을 맞이해 최근 산업 및 기술 영역에서는 인공지능을 이용한 생산력 향상, 자동화 등 딥러닝의 보편화가 빠르게 진행되고 있다. 또한, 딥러닝의 성능을 도출하기 위해서는 수많은 양의 학습용 데이터가 필요하며 그 데이터의 양은 딥러닝 모델의 성능과 정비례한다. 이에 본 작품은 최신형 영상처리 Library인 Albumentations를 이용하여 영상처리 알고리즘을 이용하여 이미지를 증강하고, 이미지 데이터 크롤링 기능을 통해 Web에서 영상 데이터를 수집을 자동화하며, Label Pix를 연동하여 수집한 데이터를 라벨링 한다. 더 나아가 라벨링 된 데이터의 증강까지 포함하여 다양한 증강 자동화를 한 인터페이스에 집적시켜 딥러닝 모델을 생성할 때 데이터 수집과 전처리를 수월하게 한다. 또한, Neural Net 기반의 AdaIN Transfer를 이용하여 이미지를 개별적으로 학습하지 않고 Real time으로 이미지의 스타일을 옮겨올 수 있도록 하여 그림 데이터의 부족 현상을 해결한다.

### 1. 서론

딥러닝의 대표적인 방법이고, 지도학습으로 객체 인식을 할 때 학습시켜야 할 매개변수 값이 적은 CNN은 데이터로부터 자동으로 특징(features)을 학습하는 대표적인 모델이다. 인간의 시각(vision) 정보 처리 방식을 흉내 낸 것으로, 특히 이미지 인식과 분류에서 탁월한 성능을 낸다. 하지만 CNN은 풀링(Pooling)을 통해 위치(translation)와 관계없이 객체를 동일하게 인식하지만, 방향(orientation)이나 비율이 달라지면 서로 다른 객체로 인식한다. 아울러 물체를 바라보는 시점 변화에 유독 취약하며 딥러닝으로 모델을 학습시킬 때, 항상 Overfitting을 고려해야 한다.

충분한 학습 데이터가 확보된다면 모델의 성능을 저해하는 Overfitting 문제가 덜 발생할 것이고, 다양한 방향, 비율을 고려하여 학습시킬 수 있지만, 양질의 데이터를 대량 확보하고, 이를 라벨링 하는 데 큰 비용과 시간이 걸린다. 또한, GAN으로 특정 스타일의 그림을 생성하도록 학습을 원할 때 그림 데이터 특성상 특정 스타일을 가진 그림의 데이터 수는 극히 한정적이다.

### 2. 개발 목적

4차 산업혁명 시대를 맞이하여 딥러닝에 관한 관심이 뜨거워지며 많은 학생들이 딥러닝을 공부하고 자신만의 모델을 구성해보려 하지만, 학습용 데이터 마련이 쉽지 않아 MNIST, CIFAT10과 같은 Open Dataset 실습 단계에서 그치는 경우가 많다.

본 시스템에서는 딥러닝으로 이미지 데이터를 학습할 때 누구나 만

드시 거쳐야 할 데이터 수집과 전처리, 증강과정을 자동화시킨 기능을 하나의 인터페이스에 집적시켜 사용자로 하여 학습용 데이터 처리의 편의성을 제공하고 소요시간을 단축해 진입장벽을 낮추고자 한다.

### 3. 본론

본 시스템은 이미지 데이터를 웹 페이지에서 크롤링해 최대한 데이터를 수집하고 다양한 증강기법을 적용한 값이지만 다른 이미지들을 생성해 데이터의 양을 극대화해 모델의 성능을 증가시킬 것이다. 하나의 인터페이스 내에서 수집한 데이터를 직접 라벨링을 할 수 있으며 YOLO 버전의 txt, Pascal VOC 버전의 xml 형식의 라벨 파일을 지원하며 바운딩박스의 정보 또한, 증강이 가능하다.

### 4. 시스템 구성요소

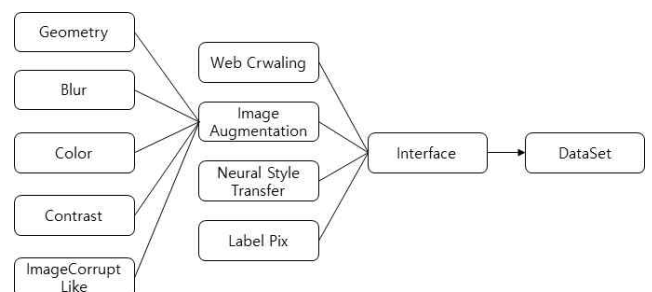


그림1. 시스템 구성도

Figure 1. System configuration diagram.

### 4-1. Alumentations 라이브러리 기반의 이미지 증강

보편화된 방법이 많은 이미지 증강은 매우 간단해 보인다. 미러링, 자르기, 밝기 및 대비 변경 등과 같은 기본 변형을 이미지에 적용하는 방법이 있고 이런 간단한 이미지 transform을 할 수 있는 많은 library들이 존재한다. 또한 Keras Image Data Generator를 통해 학습용 데이터셋 호출단계에서 데이터 증강도 가능하다. 다만, 이러한 방법에는 사소한 문제가 몇 가지 있다. 이미지 증강을 처리할 때 해당 데이터가 가진 label을 함께 증강해야 한다. 그림 2는 이미지 증강을 위한 전용 라이브러리인 Alumentations를 이용한 이미지의 Geometry 변환에 맞춰 함께 변화된 label과 mask의 모습이다.

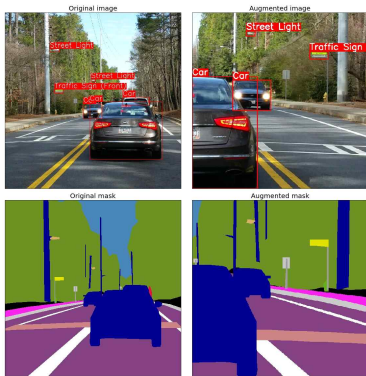


그림 2. Bounding Box, Mask가 함께 증강된 영상

Figure 2, an augmented video of Bounding Box and Mask.

동일한 이미지에 Affine transform을 적용시켰을 때 Semantic segmentation-용 mask와 Bounding Box가 변환된 이미지의 크기에 맞게 같이 따라서 변환된 것을 볼 수 있다. 따라서 label들도 같이 증강 시킴으로써 학습 가능한 데이터의 양 자체를 늘리는 효과를 볼 수 있고 회전과 조명 등 다른 noise나 변화에 대해서도 학습 시킬 수 있기에 이러한 전용 library를 사용해야 한다.

	alumentations 0.4.5	imgaug 0.4.0	torchvision (Pillow-SIMD backend) 0.5.0	keras 2.3.1	augmentor 0.2.8	solt 0.1.8
HorizontalFlip	3066	1544	1652	874	1658	853
VerticalFlip	4159	2014	1427	4147	1448	3788
Rotate	417	327	160	29	60	113
ShiftScaleRotate	703	471	144	30	-	-
Brightness	2210	997	397	210	396	2058
Contrast	2208	1023	330	-	331	2059
BrightnessContrast	2199	582	190	-	190	1051
ShiftRGB	2215	998	-	378	-	-
ShiftHSV	361	241	59	-	-	128
Gamma	2340	-	686	-	-	951
Grayscale	4961	372	735	-	1423	4286
RandomCrop64	157376	2560	41448	-	36036	35454
PadToSize512	2833	-	478	-	-	2629
Resize512	952	595	885	-	873	881
RandomSizedCrop_64_512	3128	681	1295	-	1254	2678
Equalize	760	399	-	-	666	-
Multiply	2184	1059	-	-	-	-
MultiplyElementwise	124	197	-	-	-	-

그림 3. 라이브러리별 영상처리 속도 CPU 벤치마킹 결과

Figure 3, CPU benchmarking results for video processing speed by library.

그림 3에서 나와 있듯이 Alumentations 라이브러리를 이용했을 때 타 라이브러리보다 월등한 성능점수(연산속도)를 보이기 때문에 증강 기법은 Alumentations에서 제공하는 기능을 이용했다.

### 4-2. 딥러닝 기반의 Style Transfer

Instance Normalization(IN) 방식이 기반의 AdaIN Transfer는 feature statistics를 정규화함으로써 style normalization을 수행한다는 것이다. 이는 feature statistics를 정규화하는 것이 style information을 유도할 수 있다는 기존의 연구들로 비롯된 것이다. content image와 style image를 네트워크에 통과시킬 때 나온 각각의 feature map을 저장하고, 새롭게 합성될 영상의 feature map이 content image와 style image로부터 나온 feature map과 비슷한 특성을 가지도록 영상을 최적화한다. 식 (1)에서 Batch Normalization과 마찬가지로  $\gamma, \beta$  는 학습되어야 하는 파라미터다.

Conditional Instance Normalization (CIN) 방식은 식 (4)와 같이 affine parameter인  $\gamma, \beta$  하나만 학습시키는 대신에, 각각의 style s에 따라서 서로다른 파라미터인  $\gamma_s, \beta_s$ 를 학습시키는 방법이다.

핵심 아이디어인 Adaptive Instance Normalization은 CIN으로 Feature statistics를 변경하는 것만으로 Style Transfer의 효과를 낼 수 있음을 바탕으로 미리 고정된 style을 사용하지 말고 임의의 (arbitrary) 이미지로부터 스타일을 가져오자는 아이디어로 IN을 간단하게 확장한 AdaIN이 되었다. 식 (5)와 같이 AdaIN을 이용하면 원하는 이미지에서 Style을 가져와 적용할 수 있다. AdaIN은 channel-wise mean and variance라는 feature statistics를 transferring 함으로서 feature space 상에서 style transfer를 수행한다. 여기서  $x, y$ 는 각각 Content image와 Style image를 나타낸다.



그림 4. AdaIN Transfer 적용 결과

Figure 4. AdaIN Transfer application results.

$$IN(x) = \gamma \left( \frac{x - \mu(x)}{\sigma(x)} \right) + \beta \quad (1)$$

$$\mu_{nc}(x) = \frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W x_{nchw} \quad (2)$$

$$\sigma_{nc}(x) = \sqrt{\frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W (x_{nchw} - \mu_{nc}(x))^2 + \epsilon} \quad (3)$$

$$CIN(x; s) = \gamma^s \left( \frac{x - \mu(x)}{\sigma(x)} \right) + \beta^s \quad (4)$$

$$AdaIN(x, y) = \text{signa}(y) \left( \frac{x - \mu(x)}{\sigma(x)} \right) + \mu(y) \quad (5)$$

### 4-3. Web Image Crawling

그림 5는 Selenium 라이브러리를 이용하여 크롬 웹 브라우저를 자동으로 제어하여 원하는 키워드의 이미지를 검색해서 키워드에 해당하는 이미지들을 지정한 저장 경로에 자동으로 저장하는 알고리즘을 실행한 화면이다. 네이버 이미지검색을 기반으로 최상단의 이미지부터 크롤링을 시작해 지정한 개수에 도달하면 크롤링을 종료한다.



그림 5. 네이버 이미지 크롤링으로 데이터가 저장되는 과정  
Figure 5. The process of storing data through Naver image crawling.

### 4-4. Label Pix

Label Pix를 GUI에 연동하여 크롤링하거나 Input 한 데이터를 직접 바운딩박스를 그려 라벨링할 수 있으며 라벨링된 데이터는 xml형식, YOLO버전 txt 형식으로 저장이 가능하다. 또한, 이미지 증강을 거칠 때 바운딩박스 정보가 함께 변환되어 증강된다.

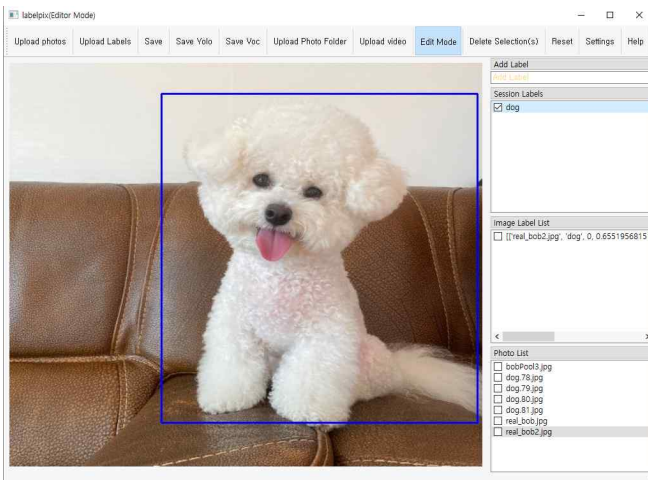


그림 6. LabelPix를 이용한 데이터 라벨링  
Figure 6. Labeling data using LabelPix.

### 4-5. Interface

그림 7, 그림 8은 PyQt5 라이브러리를 이용하여 이미지 증강, 데이터 크롤링, 데이터 라벨링, Style Transfer의 기능을 하나의 interface에 집적시켜 기능들을 사용할 수 있도록 구현한 화면이다. 순서는 다음과 같다.

#### <Image Augmentaion>

1. Input data 지정
2. 적용할 증강 기법 체크한후 매개변수 조정
3. 미리보기 버튼을 통해 출력될 이미지를 확인 후 매개변수 재조정
4. 증강된 이미지가 저장될 경로를 설정 후 증강.

#### <Data Crawling>

1. Crwaling / Style Transfer 탭에서 검색 키워드와 개수 입력
2. 크롤링된 이미지 저장할 경로 설정 후 크롤링

#### <Style Transfer>

1. Crwaling / Style Transfer 탭에서 Content 이미지와 Style 이미지 지정.
2. 미리보기 버튼을 이용해 증강 결과를 보며 alpha 값을 조정.
3. 증강된 이미지가 저장될 경로를 설정 후 증강.

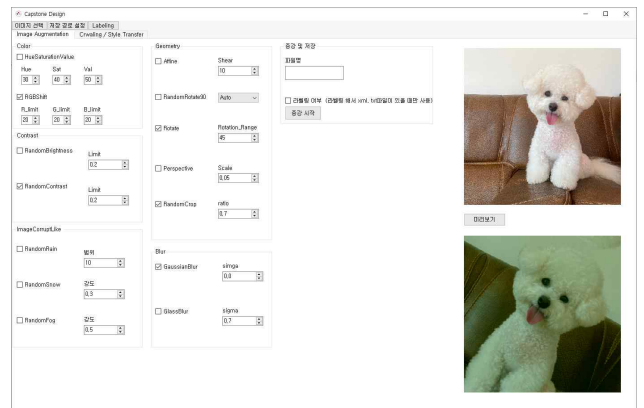


Fig 7. Image Augmentation Interface Tab

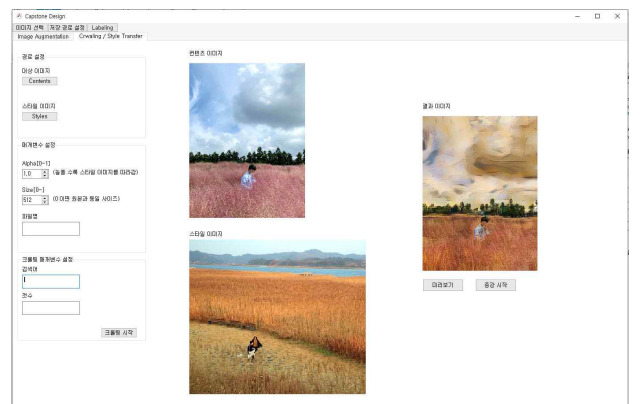


Fig 8. Crawling / Style Transfer Tab

## 5. 실험 및 결과

### 5-1. 실험 데이터셋

본 시스템에서는 Kaggle의 Cat&Dog 데이터셋을 이용하여 데이터 증강에 따른 정확도 향상의 유효성을 확인하기 위한 실험을 진행하였다. 실험을 위하여 train, validation, test 데이터셋을 각각 20,000, 4,000, 1,000개로 구성하였으며 Tensorflow의 Keras 라이브러리를 이용하여 ResNet-50V2 전이학습 모델을 이용했다. CNN 계층만 ResNet-50V2 모델을 이용하였고 Fully Connected Layer를

```
Dense(256, activation='relu'),
Dropout(0.5),
Dense(128, activation='relu'),
Dropout(0.25),
```

Dense(1,activation='sigmoid') 으로 구성하였으며 하나와 최적화 알고리즘으로는 Stochastic Gradient Descent (SGD)를 사용하였다. 실험은 Back born을 Freezing하고 학습한 model, Freezing을 해제한 후 학습한 model, Freezing을 해제하고 각 클래스당 1000장씩 이미지를 크롤링을 통해 데이터를 추가로 확보하고 이미지 증강을 거친 model의 정확도를 비교하는 것으로 이뤄진다.

### 5-2. 실험 결과 및 결론

크롤링을 통해 클래스별로 1,000장의 데이터를 확보하고 데이터 증강을 거친 Back born freezing을 해제한 모델에서 데이터 증강 전 후의 0.5%의 성능 향상을 보였다. 따라서 본 시스템을 통해서 데이터를 수집하고 증강함에 따라서 딥러닝 모델의 성능이 향상될 수 있음을 확인할 수 있다. 이를 통해서 딥러닝 학습자가 신경망을 구현할 수 있다면, 데이터 수집과 전처리 과정의 진입장벽을 본 시스템을 통해 낮출 수 있음을 보였다.

향후 과제로는 머신러닝을 통해서 비슷한 클래스별로 데이터를 일정 수준 오토라벨링 하는 성능을 추가하여 라벨링에 대한 자동화 기능을 추가할 필요가 있으며, 네이버 홈페이지 뿐만 아니라 다양한 웹 페이지 상에서 이미지를 크롤링할 수 있도록 기능을 보강하도록 해야 한다.

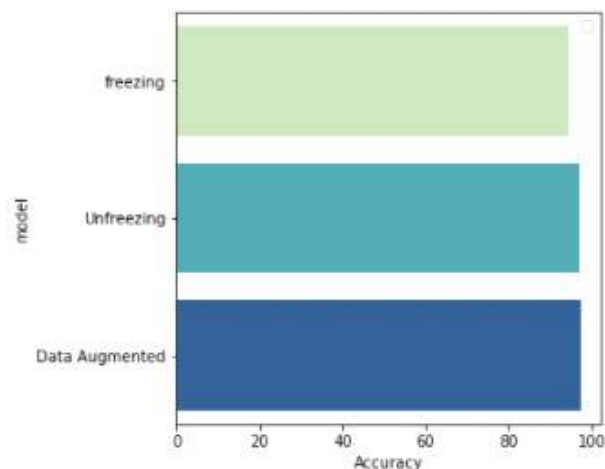


그림 9. 각 model의 정확도 결과 비교

Figure 9. Comparison of the accuracy results of each model.

	Freezing	Unfreezing	Augmented Unfreezing Data
Accuracy	94.5%	97%	97.5%

표 1. 각 모델의 정확도 결과 수치

Table 1. Accuracy result figures for each model.

## 참 고 문 헌

- [1] Xun Huang, Serge Belongie, "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization", ICCV, (Mar 2017)
- [2] Alexander Buslaev, Alex Parinov, Eugene Khvedchenya, Vladimir I. Iglovikov, Alexandr A. Kalinin, "Albumentations: fast and flexible image augmentations", Information, (Sep 2018)
- [3] Tamar Rott Shaham, Tali Dekel, Tomer Michaeli, "SinGAN: Learning a Generative Model from a Single Natural Image", ICCV, (Sep 2019)