



# “나만의 반려동물 이모티콘”

한보혜, 유광열, 김홍일, 정승철

**부엉이들 재워조**

# Index



아이디어



데이터 수집 및 전처리



구현 과정



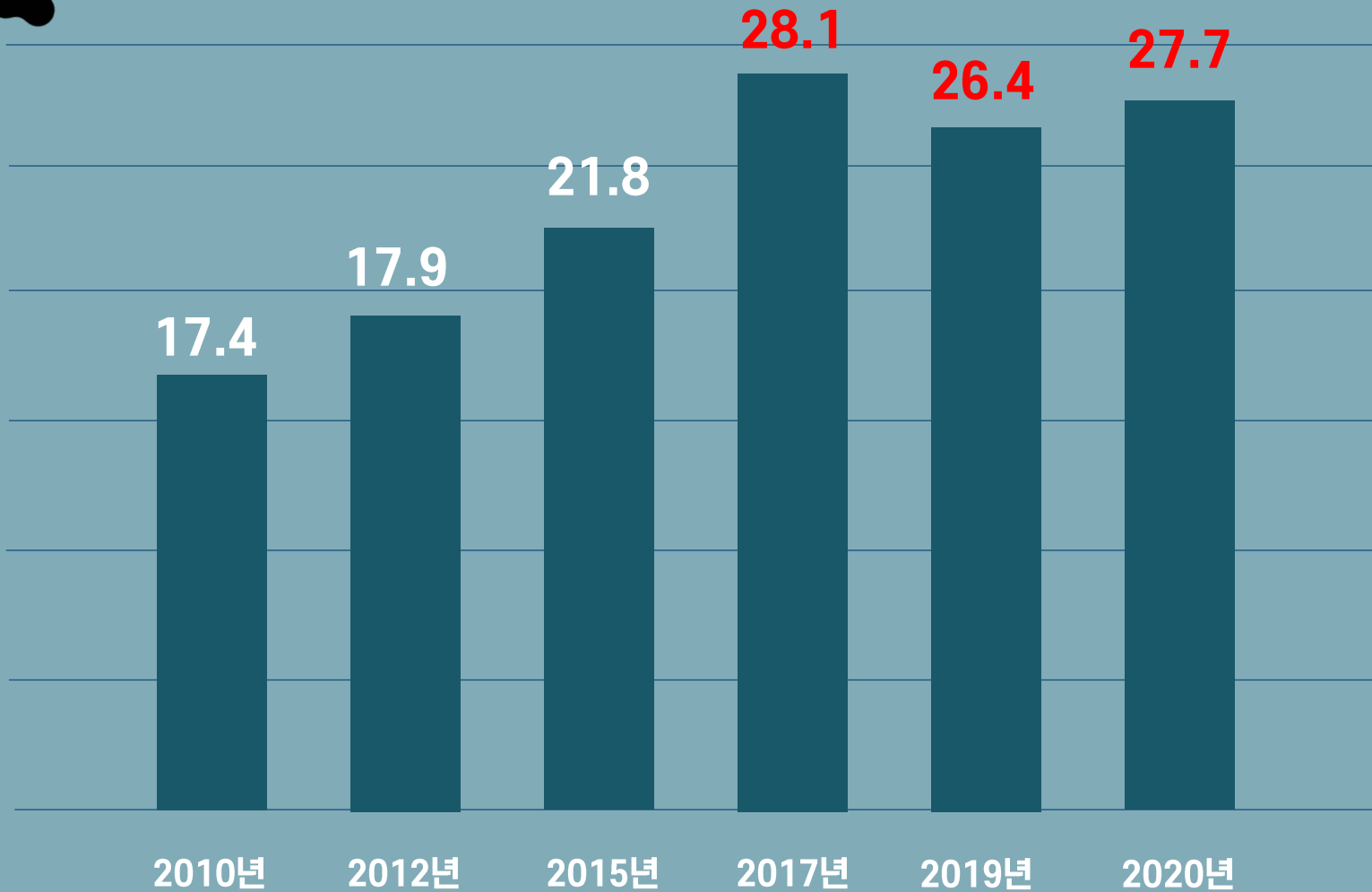
결과 및 개선 방향



## 01. 아이디어

## 우리나라 반려동물 양육가구 비율

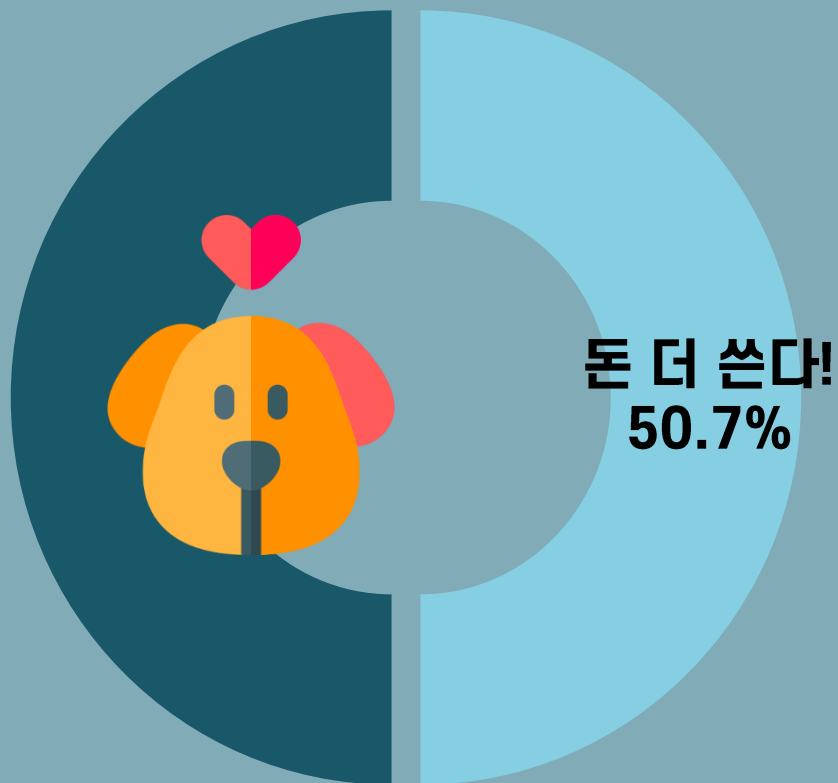
자료 : 농식품부



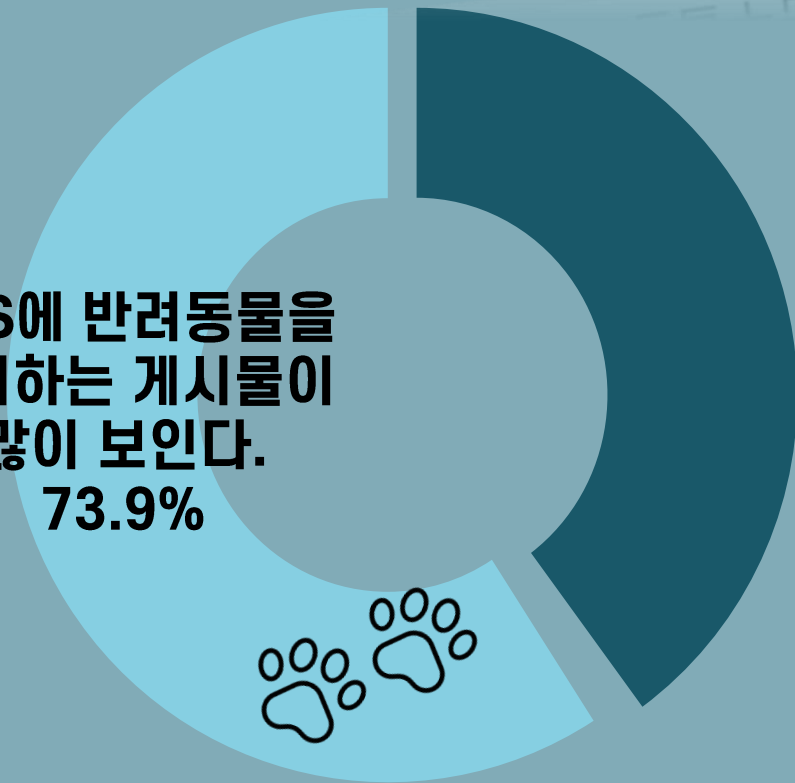
[인포그래픽] 반려동물은 이제 가족 "돈 아깝지 않아요"

펫팸족 절반, 자신보다  
'반려동물'에게 돈 더 쓴다

[HI★PICK] 태연·키·가비... 스타만큼 유명한 '견플루언서'



SNS에 반려동물을  
과시하는 게시물이  
많이 보인다.  
73.9%



\*자료:엠브레인 트렌드모니터



## IDEA의 시작



+



=

?

나의 반려동물이 내가 원하는 모양의 이모티콘으로 나올 수 있다면…?



## 02. 데이터 수집 및 전처리



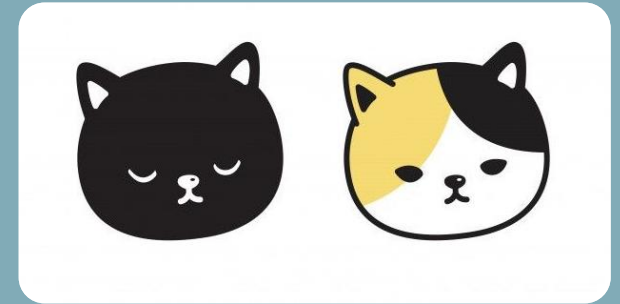
# 이모티콘 데이터 수집



총 70,828 개의 데이터



총 46,845 개의 데이터



총 216 개의 데이터





## 실제 강아지,고양이 데이터 수집



총 3,239 개의 데이터



총 3,499 개의 데이터



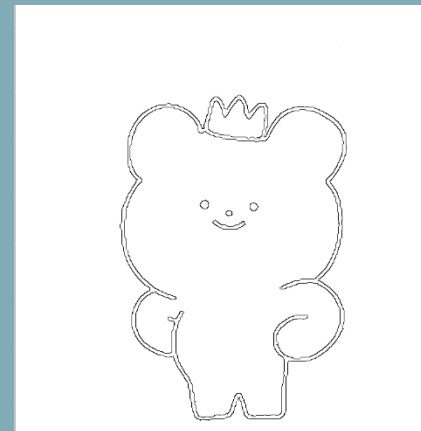
# 데이터 전처리



배경 생성



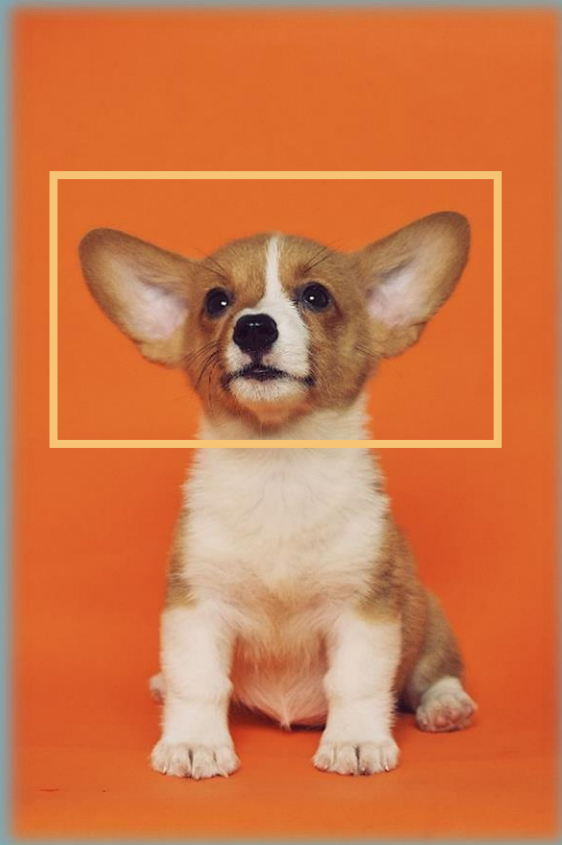
텍스트 제거



도면 생성



## 데이터 전처리



귀와 얼굴이 잘리지 않은  
216개의 데이터 확보



## 03. 구현 과정



## 사용된 모델



**CartoonGAN**



**Pix2Pix**

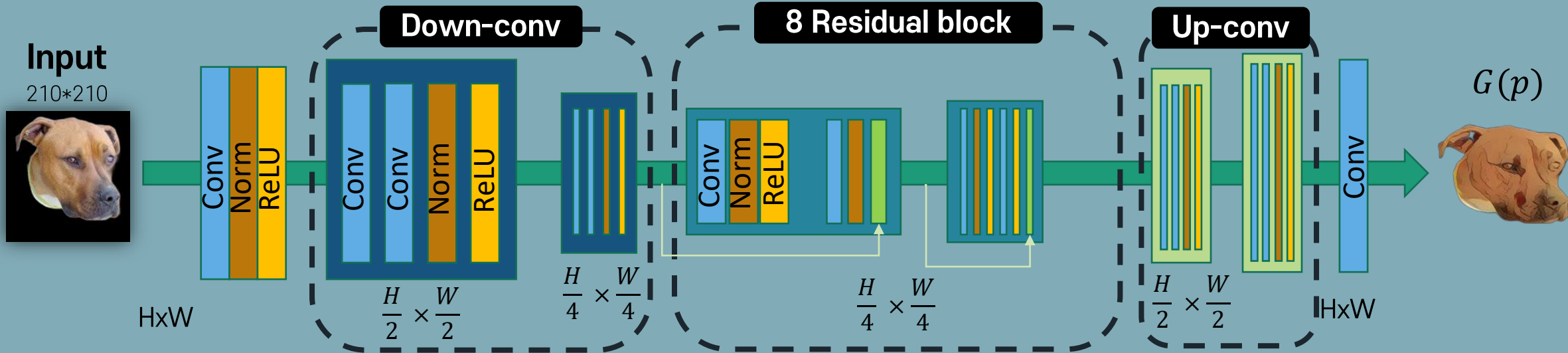


**StyleTransfer**

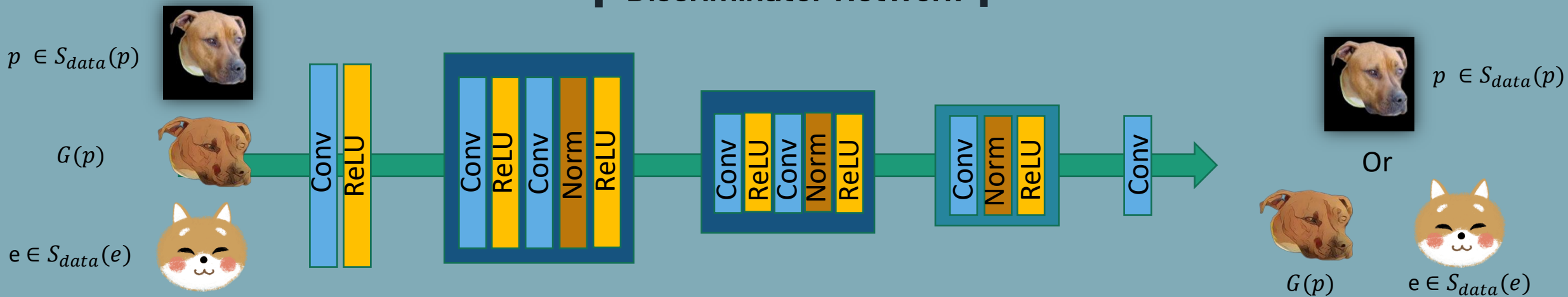


# CartoonGAN

## Generator Network



## Discriminator Network





# CartoonGAN

## CartoonGAN Loss

$$\mathcal{L}(G, D) = \mathcal{L}_{adv}(G, D) + \omega \mathcal{L}_{con}(G, D)$$

$$\begin{aligned} \mathcal{L}_{adv}(G, D) = & \mathbb{E}_{c_i \sim S_{data}(c)} [\log D(c_i)] \\ & + \mathbb{E}_{e_j \sim S_{data}(e)} [\log(1 - D(e_j))] \\ & + \mathbb{E}_{p_k \sim S_{data}(p)} [\log(1 - D(G(p_k)))]. \end{aligned}$$

Cartoon image  $\rightarrow 1$

Edge smoothed  $\rightarrow 0$

Generated Cartoon image  $\rightarrow 0$

### → Adversarial Loss

이모지의 특성을 반영하여  
“Edge-Smoothing”이 추가되었다.

Canny Edge 검출 – 영역확장 – Gaussian Smoothing

$$\mathcal{L}(G, D) = \mathcal{L}_{adv}(G, D) + \omega \mathcal{L}_{con}(G, D)$$

### → Content Loss

Input 사진의 내용을 유지하도록 강제한다.  
Content Loss의 영향은  $w$ 를 통해 조절된다.

$W=10$ 을 사용



# CartoonGAN

**1** Batch Norm



Instance Norm

이미지 contrast에 의존도가 높음을 개선하기 위함.

**2** ReLU



LeakyReLU

학습의 Stability를 높이기 위함.

**3** VGG



ResNet

Feature vector 추출 성능을 개선하기 위함.





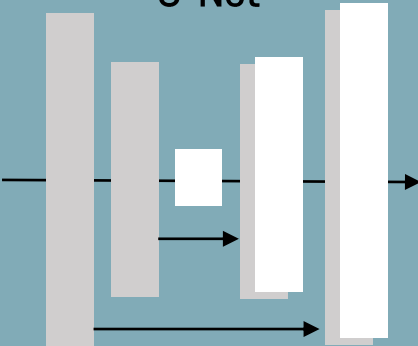
# Pix2Pix

## Generator Network

Input



U-Net



$G(p)$



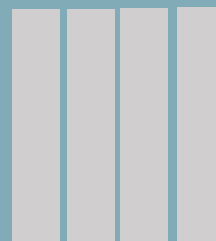
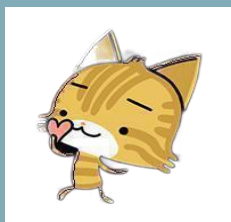
## Discriminator Network



+

Real

$G(p)$



PatchGAN

Real / Fake



# Pix2Pix

## Pix2Pix Loss

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$

### → Adversarial Loss

BCE Loss를 사용한  
우리에게 익숙한 기본 GAN Loss이다.

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$

### → L1 Loss

이미지가 흐리게 나오는 문제를 개선하기 위해  
L1 distance를 사용한다.

람다를 통해 최적의 loss 탐색



# Pix2Pix

①

BCE Loss



Least Square Loss

학습 안정성 개선을 위함.

②

Data Augmentation

데이터의 부족으로  
다양성을 주기 위함.

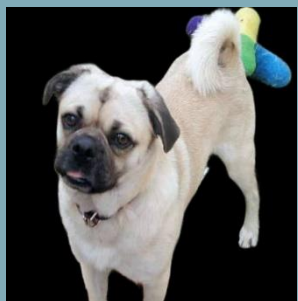
③

openCV를 통한  
사후 개선

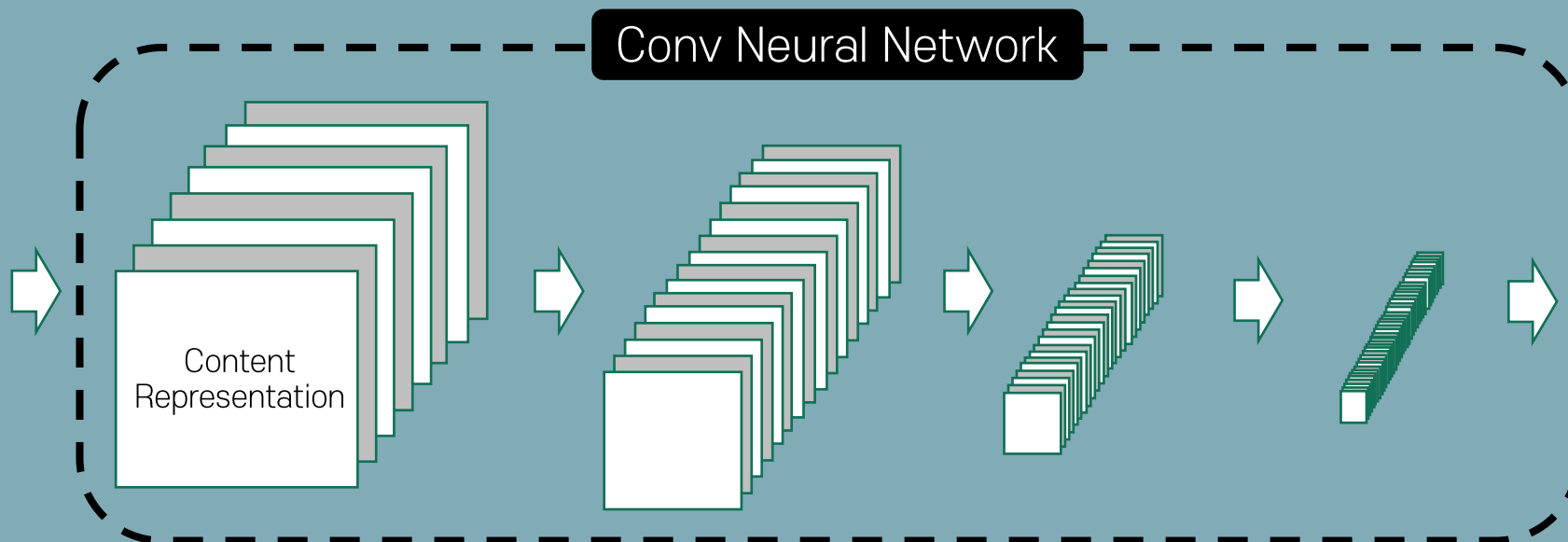
선이 흐리게 나오는  
결과를 개선하기 위함.



# StyleTransfer



Input



\* Pretrained 된 VGG16 구조의 CNN

CNN모델에 존재하는 Feature Map을 활용하여  
Content 정보를 담고 있는 이미지와 Style 정보를 담고 있는 이미지를 학습한다.



# StyleTransfer

## StyleTransfer Loss

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style}$$

$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2 .$$

### → Content Loss

P벡터 이미지의 feature map 과  
x 노이즈 벡터 이미지의 feature map이  
같아지도록 하는 Loss

$$\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^L w_l E_l,$$

### → Style Loss

Style 이미지 a의 Gram 매트릭스와  
X 벡터 이미지 의 Gram 매트릭스가  
같아지도록 하는 Loss

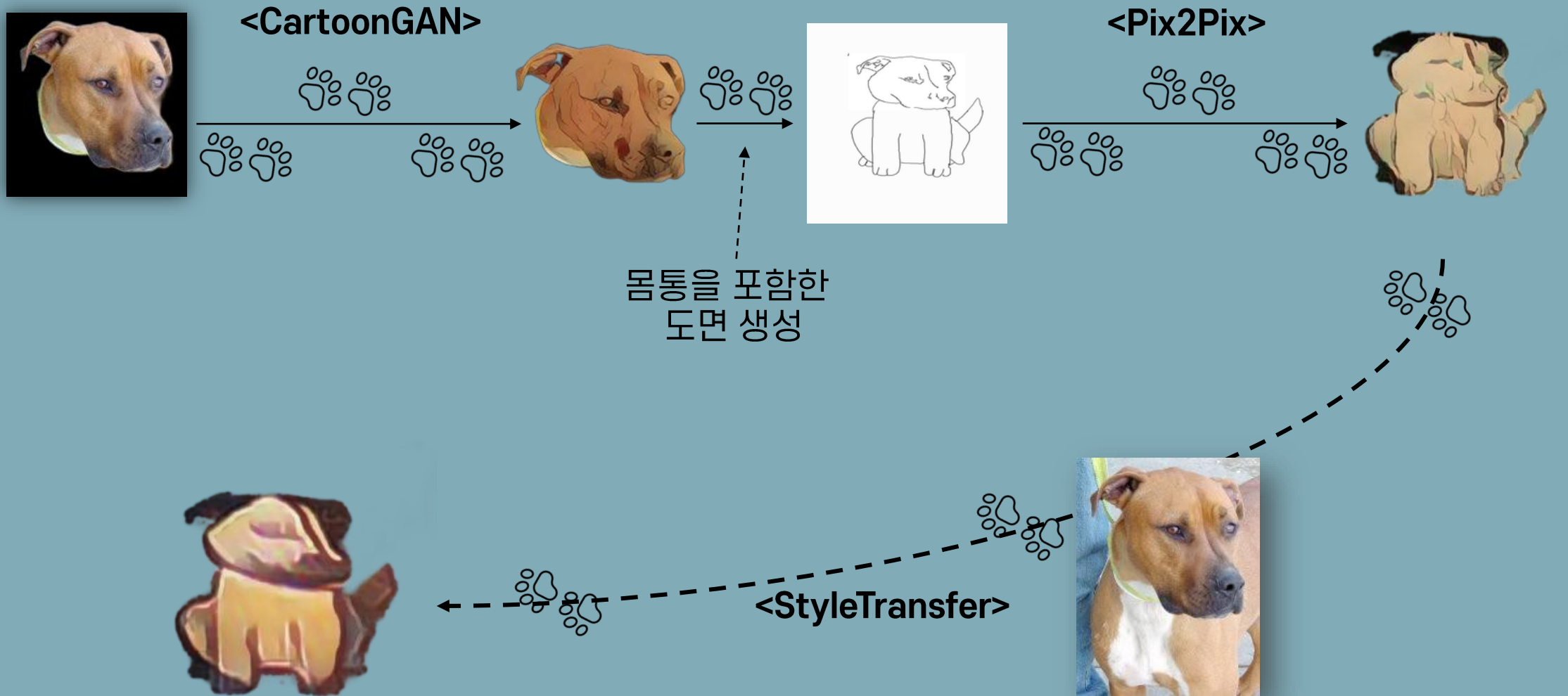
- \* Style : 서로 다른 특징 간의 상관관계
- \* Gram 매트릭스 : CNN 채널에서 나온 activation들 의 상관관계를 표현한 행렬



## **04.결과 및 개선방향**



# 결과



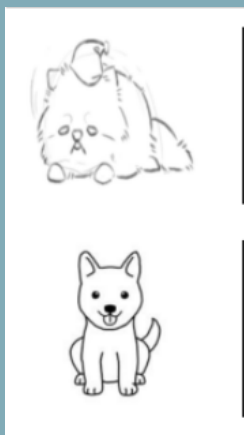


# 활용 방향

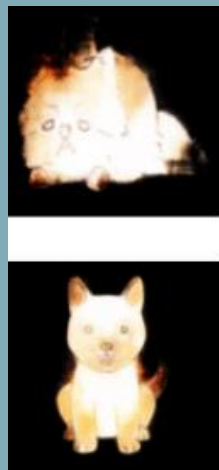
1



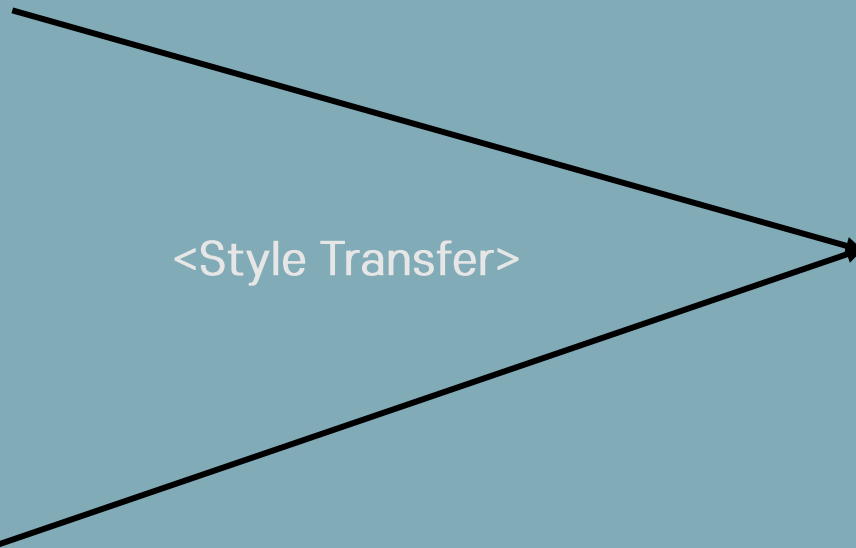
<CartoonGAN>



<Pix2Pix>



<Style Transfer>







## 활용 방향

2



도면 추출



pix2pix & style Transfer





## 어려웠던 점

- 1 이모티콘의 선이 너무 둥글고 경계가 명확하지 않았던 점.
- 2 이모티콘의 스타일이 너무 다양하고 하나의 스타일에 대한 데이터가 너무 적었던 점.
- 3 Style Transfer에 줄무늬와 얼룩무늬가 적용이 안되었던 점.
- 4 실제와 이모티콘 사이의 차이가 너무 커 오차를 줄이는 데 어려움이 있었음.
- 5 데이터 전처리가 너무 복잡하고 수작업이 너무 많았던 점.

