IMAGE-TO-IMAGE TRANSLATION WITH CONDITIONAL ADVERSARIAL NETWORKS

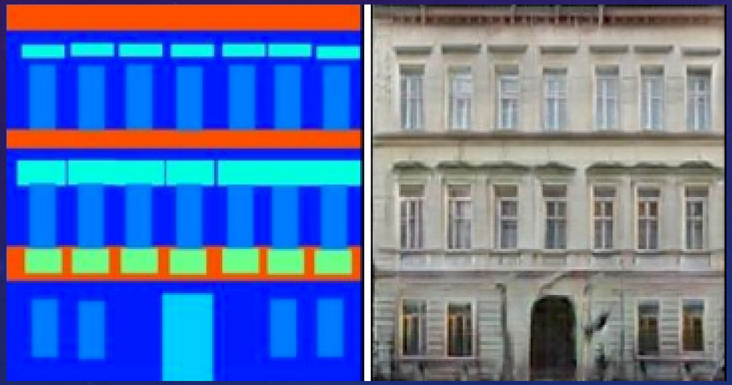
NATURE LANGUAGE PROCCESSING

TA: 顏苙峯

DATE:5/17

ABSTRACT

We investigate conditional adversarial networks as a general-purpose solution to image-to-image translation problems



CONTENT

- Introduction
- Related work
- Method
- Experiments
- Conclusion

CONTENT

- Introduction
- Related work
- Method
- Experiments
- Conclusion

INTRODUCTION

- Many problems in image processing, computer graphics, and computer vision can be posed as “translating” an input image into a corresponding output image. Just as a concept may be expressed in either English or French

INTRODUCTION

- we define automatic image-to-image translation as the task of translating one possible representation of a scene into another, given sufficient training data
- we explore GANs in the conditional setting. Just as GANs learn a generative model of data, conditional GANs learn a conditional generative model

INTRODUCTION- GOAL

- Our goal in this paper is to develop a common framework for all these problems.

CONTENT

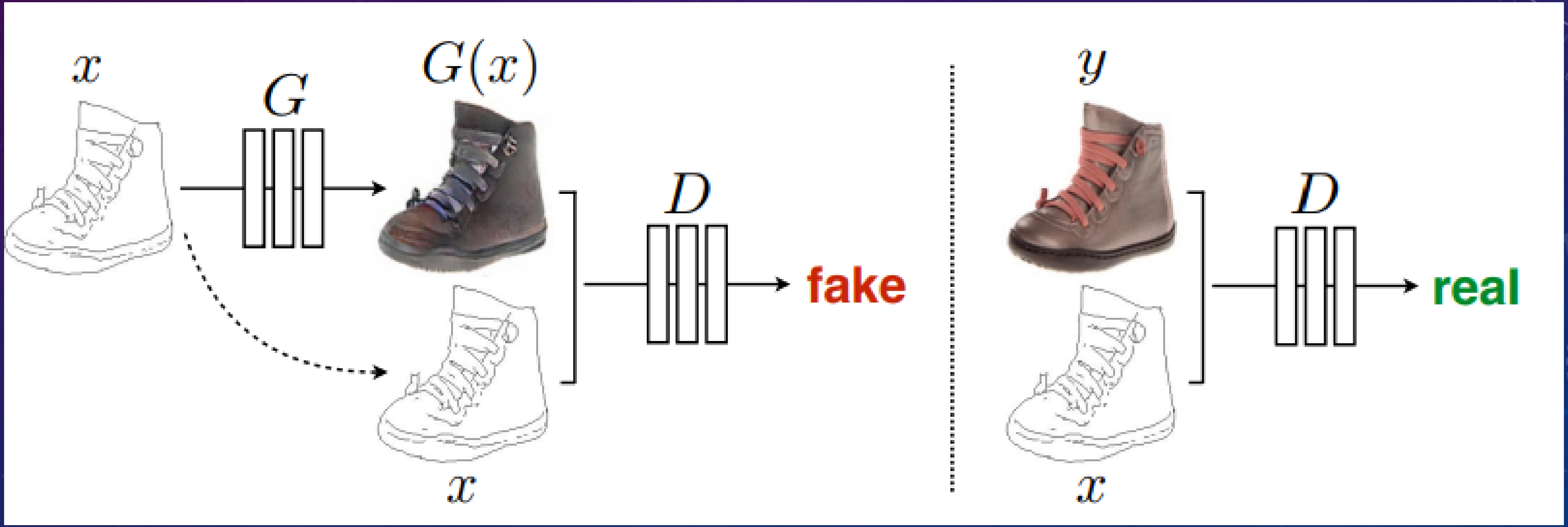
- Introduction
- Related work
- Method
- Experiments
- Conclusion

CONTENT

- Introduction
- Related work
- Method
- Experiments
- Conclusion

STRUCTURED LOSSES FOR IMAGE MODELING

- Image-to-image translation problems are often formulated as per-pixel classification or regression
- These formulations treat the output space as "unstructured" in the sense that each output pixel is considered conditionally independent from all others given the input image. Conditional GANs instead learn a structured loss. Structured losses penalize the joint configuration of the output.



x
G
$G(x)$
D
fake
x
y
D
real
x

RELATED WORK

- Our method also differs from the prior works in several architectural choices for the generator and discriminator. Unlike past work, for our generator we use a "U-Net"-based architecture
- Our discriminator we use a convolutional "PatchGAN" classifier

CONTENT

- Introduction
- Related work
- Method
- Experiments
- Conclusion

CONTENT

- Introduction
- Related work
- Method
- Experiments
- Conclusion

OBJECTIVE

- **x**: input image, **z**: noise image, **y:** real image

OBJECTIVE

- GAN $G: z \rightarrow y$
- conditional GANs $G: \{x, z\} \rightarrow y$

OBJECTIVE

Conditional GAN

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,y}[log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z))]$$

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

OBJECTIVE

$$\mathcal{L}_{GAN}(G,D)= \mathbb{E}_y[logD(y)]+ \mathbb{E}_{x,z}[\log(1-D(x,G(x,z))]$$

$$G^* = argmin_G max_D \mathcal{L}_{cGAN}(G,D)$$

- $\mathcal{L}_{\mathrm{L1}}(G) = \mathbb{E}_{x,y,z}[||y - G(x, z)||_1]$

OBJECTIVE _FINAL OBJECTIVE

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

NETWORK ARCHITECTURES

We adapt our generator and discriminator architectures

Both generator and discriminator use modules of the form

convolution-BatchNorm-ReLu

Generator and Discriminator

Generator and Discriminator

TWO ARCHITECTURE CHOICES FOR THE GENERATOR

U-NET VS ENCODER

TWO ARCHITECTURE CHOICES FOR THE GENERATOR

U-NET   VS   ENCODER

ENCODER-DECODER

the input is passed through a series of layers

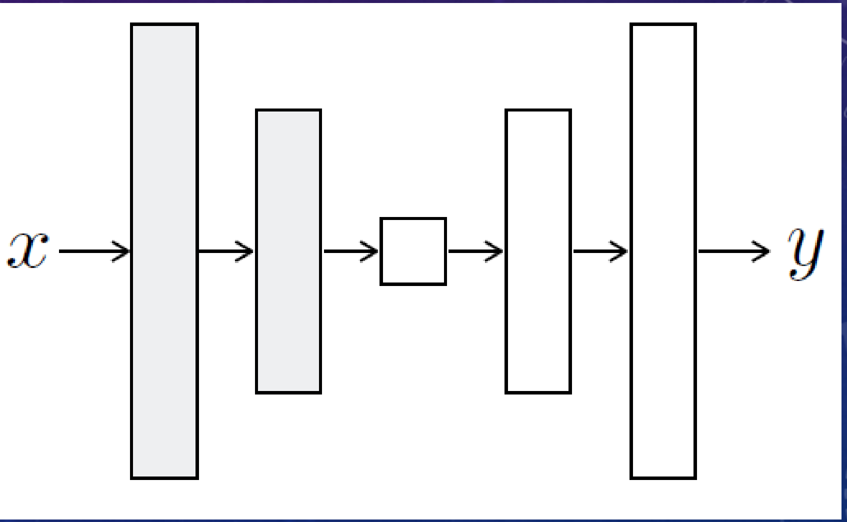
until a bottleneck layer,at which point the process is reversed

For many image translation problems,

there is a great deal of low-level information shared

between the input and output, and it would be desirable to

shuttle this information directly across the net.

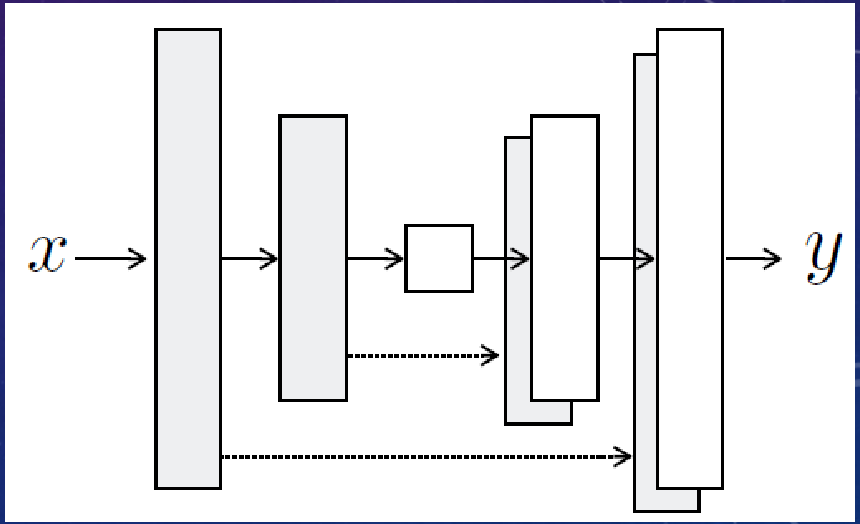


TWO ARCHITECTURE CHOICES FOR THE GENERATOR

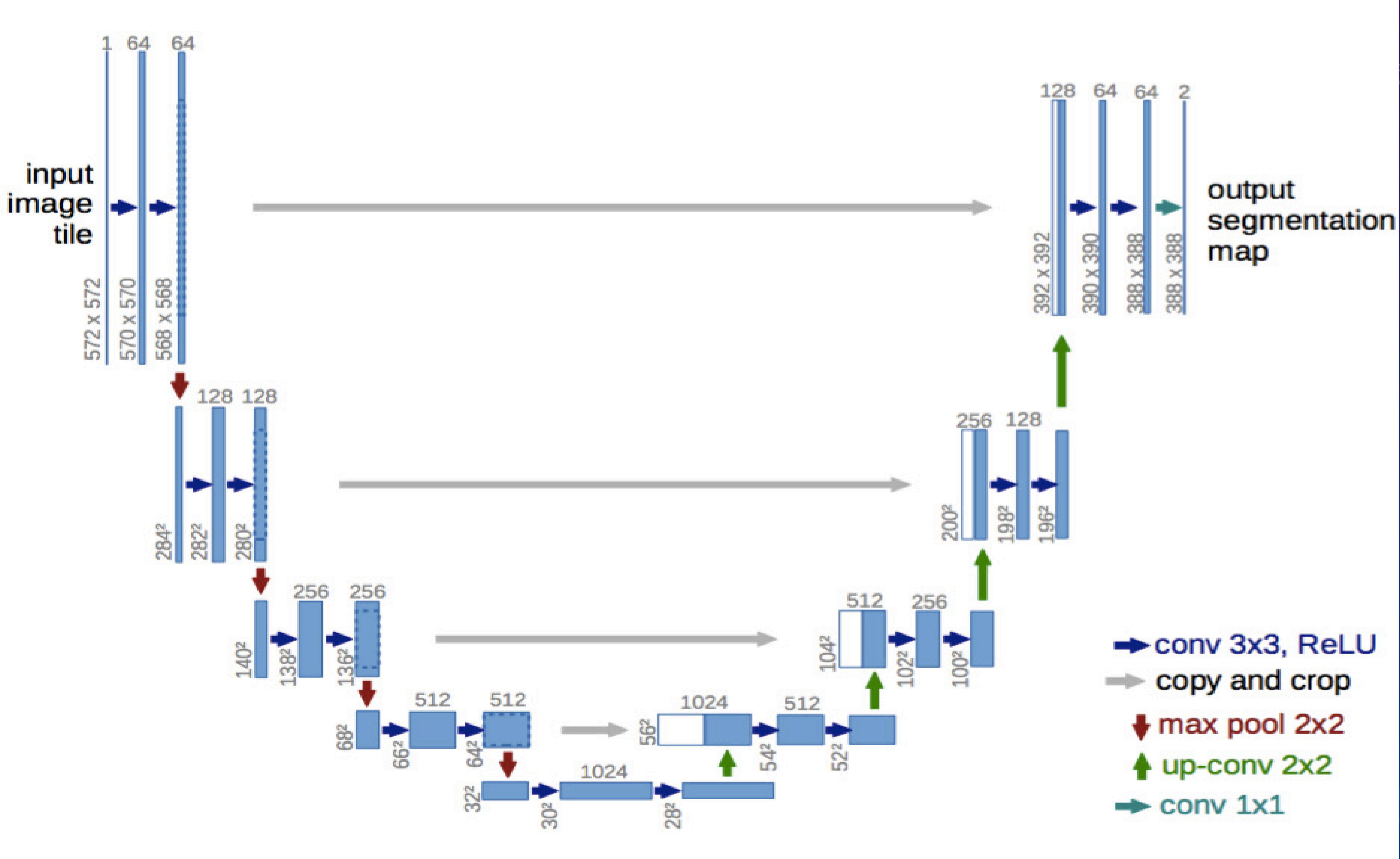
U-NET VS ENCODER

U-NET

To give the generator a means to circumvent the bottleneck for information like this, we add skip connections, following the general shape of a “U-Net” . Specifically, we add skip connections between each layer i and layer where n is the total number of layers. Each skip connection simply concatenates all channels at layer i with those at layer

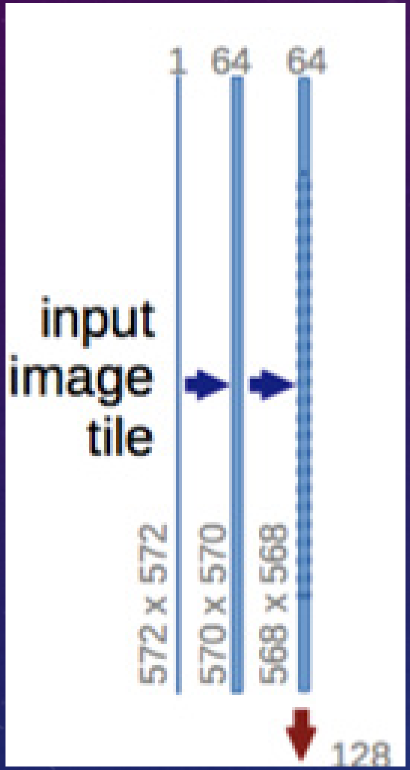


U-NET



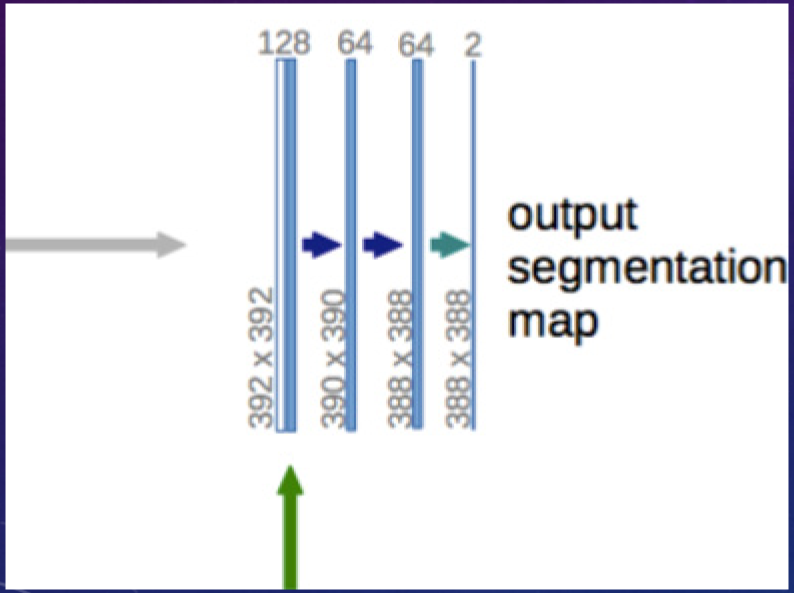
From PAPER:U-Net: Convolutional Networks for Biomedical Image Segmentation

U-NET



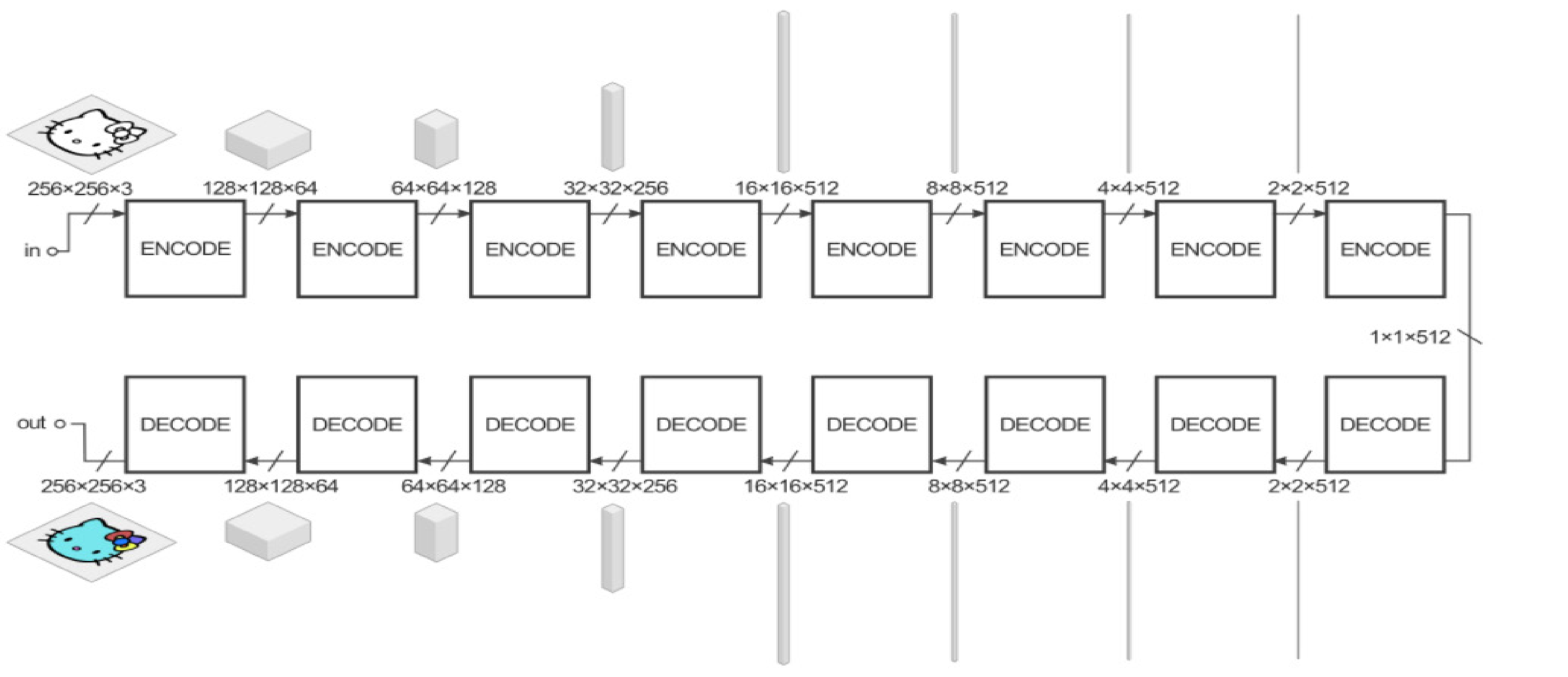

```python
down1 = Conv2D(64, (3, 3), padding='same')(inputs)
down1 = BatchNormalization()(down1)
down1 = Activation('relu')(down1)
down1 = Conv2D(64, (3, 3), padding='same')(down1)
down1 = BatchNormalization()(down1)
down1 = Activation('relu')(down1)
down1_pool = MaxPooling2D((2, 2), strides=(2, 2))(down1)
```

U-NET

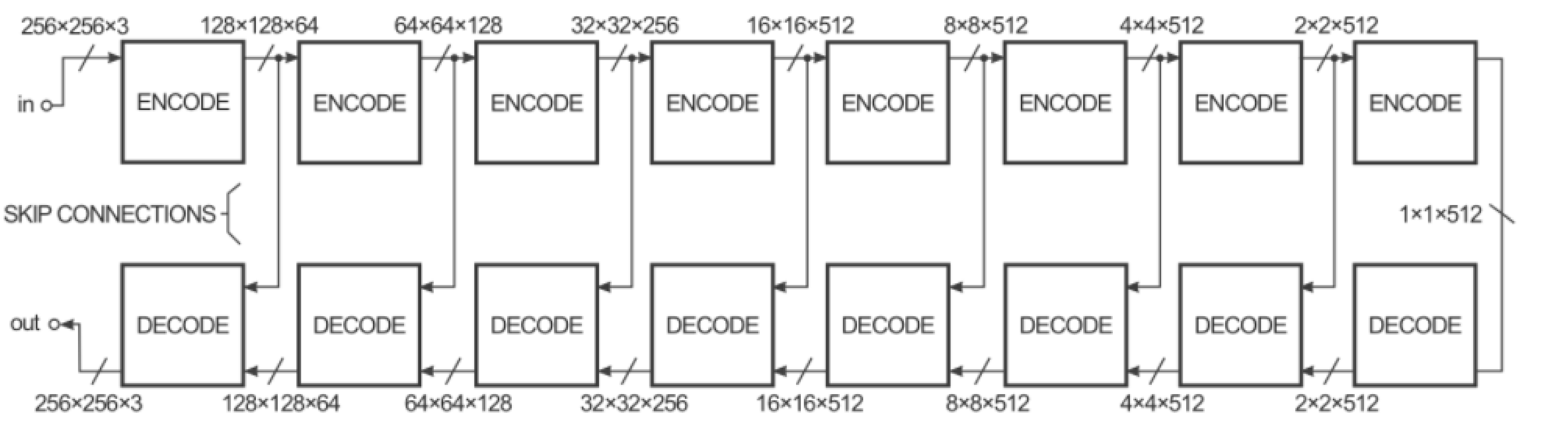



```
up1 = UpSampling2D((2, 2))(up2)
up1 = concatenate([down1, up1], axis=3)
up1 = Conv2D(64, (3, 3), padding='same')(up1)
up1 = BatchNormalization()(up1)
up1 = Activation('relu')(up1)
up1 = Conv2D(64, (3, 3), padding='same')(up1)
up1 = BatchNormalization()(up1)
up1 = Activation('relu')(up1)
up1 = Conv2D(64, (3, 3), padding='same')(up1)
up1 = BatchNormalization()(up1)
up1 = Activation('relu')(up1)
```

ENCODER-DECODER



U-NET

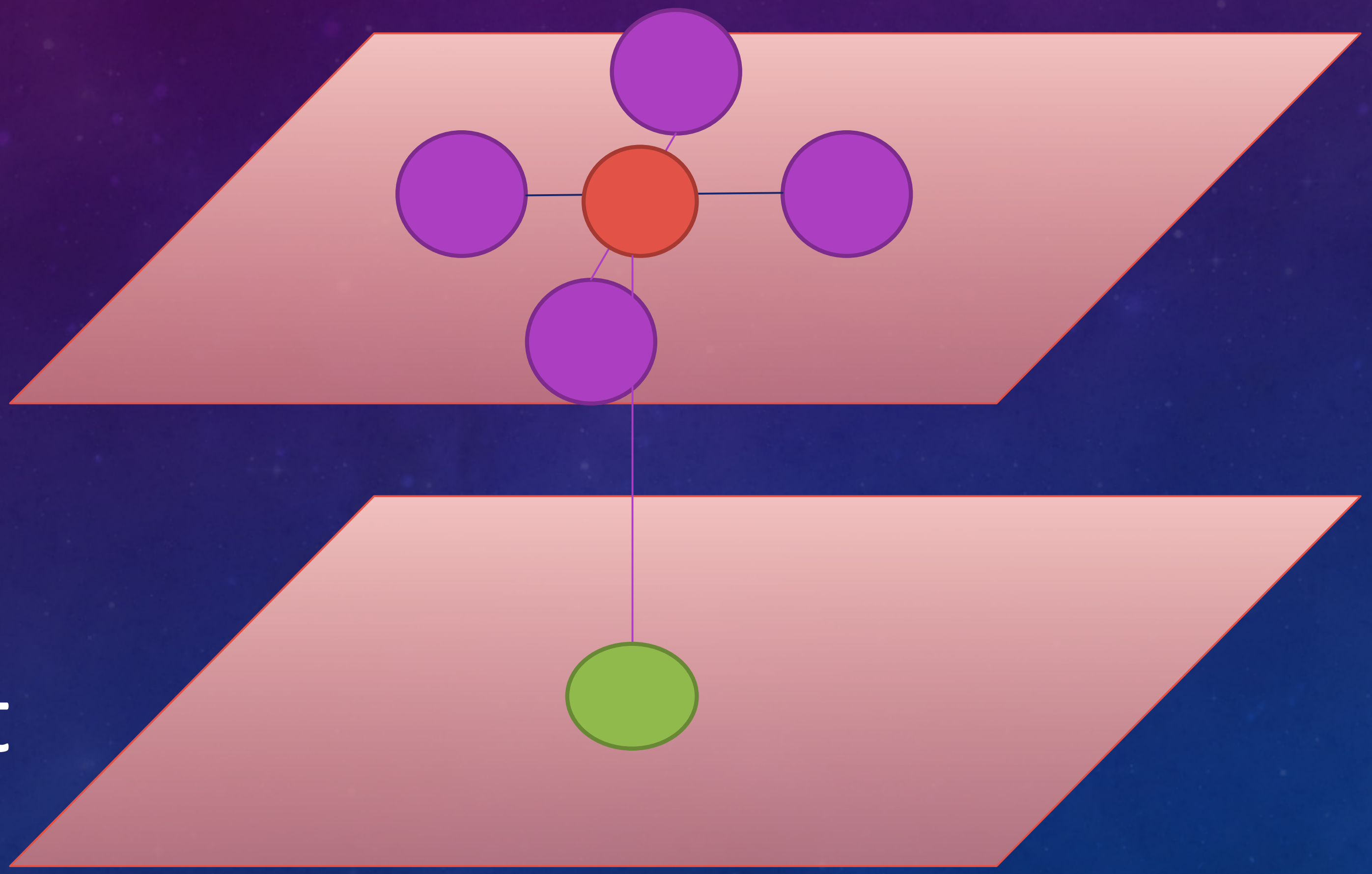

256×256×3
128×128×64
64×64×128
32×32×256
16×16×512
8×8×512
4×4×512
2×2×512
in
ENCODE
SKIP CONNECTIONS
1×1×512
out
DECODE
256×256×3
128×128×64
64×64×128
32×32×256
16×16×512
8×8×512
4×4×512
2×2×512

Generator and Discriminator

Generator and Discriminator

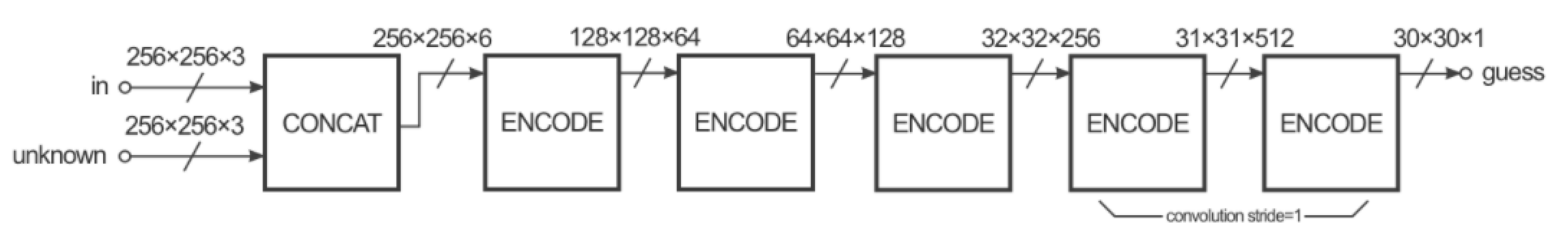
MARCOV RANDOM FIELD

Result



Input

MARKOVIAN DISCRIMINATOR



CONTENT

- Introduction
- Related work
- Method
- Experiments
- Conclusion

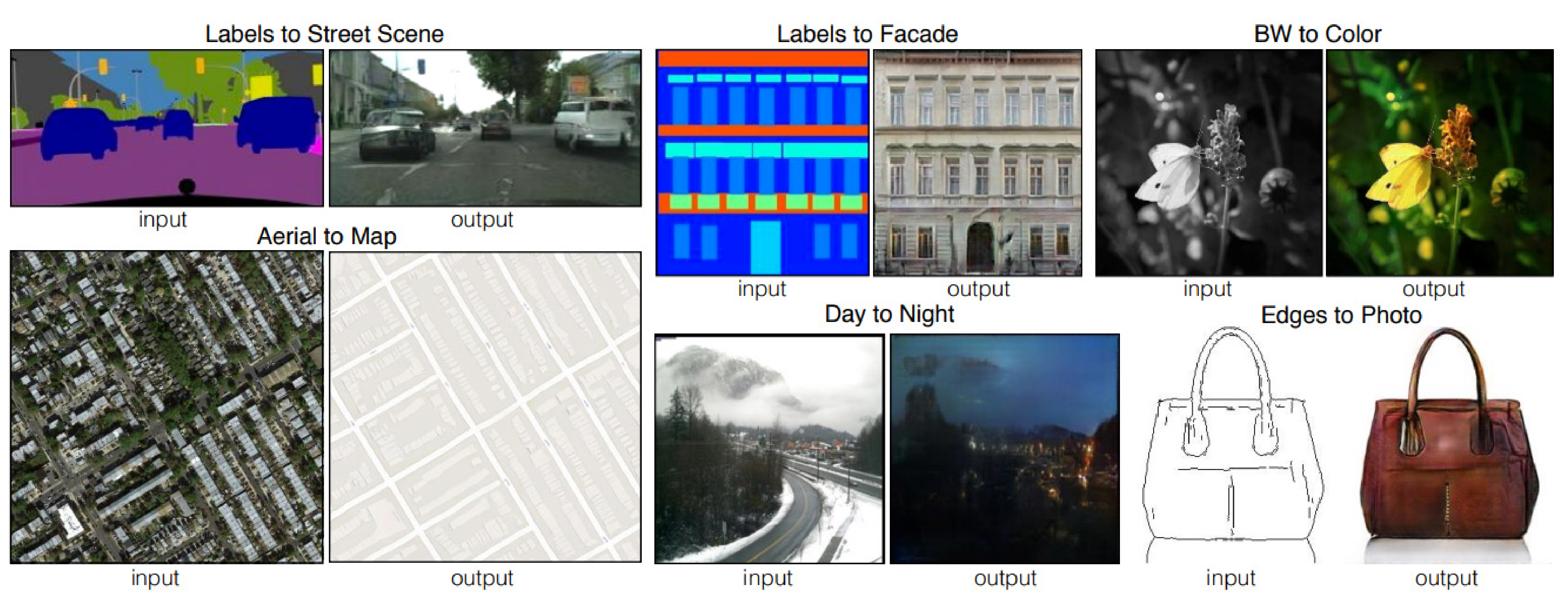
CONTENT

- Introduction
- Related work
- Method
- Experiments
- Conclusion

EXPERIMENTS

- Semantic labels->photo, trained on the Cityscapesdataset
- Architectural labels->photo, trained on CMP Facades
- Map->aerial photo, trained on data scraped from Google Maps.
- BW->color photos
- Edges->photo using the HED edge detector plus postprocessing.
- Sketch->photo: tests edges!photo models on humandrawn
- Day->night
- Photo with missing pixels->inpainted photo, trained on Paris StreetView

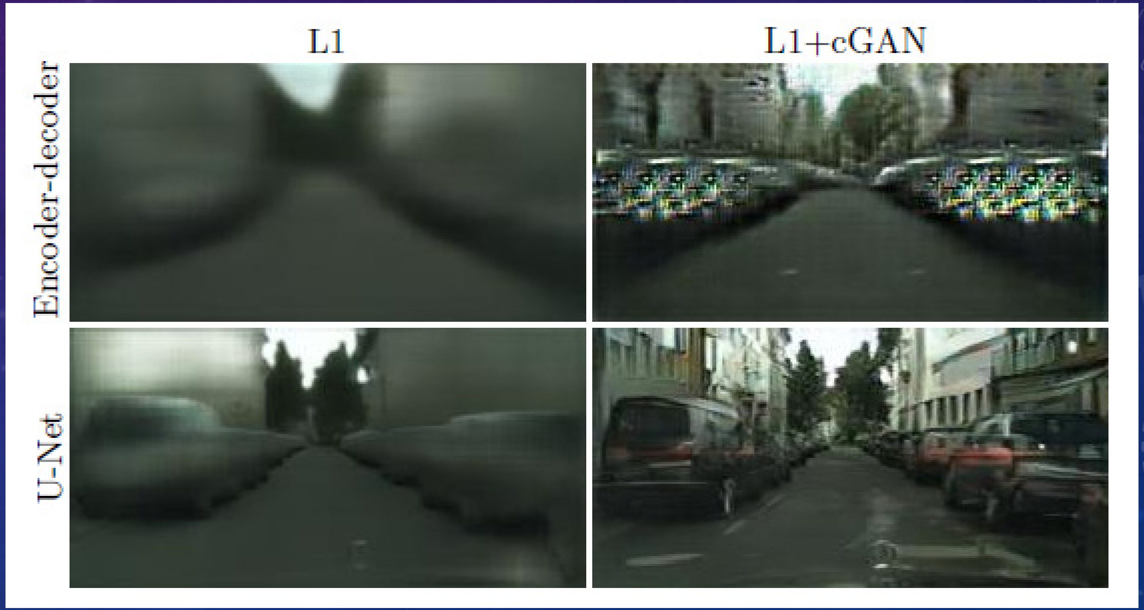
EXPERIMENTS



EXPERIMENTS

EXPERIMENTS

Adding skip connections to an encoder-decoder to create a “U-Net” results in much higher quality results.



EXPERIMENTS

Patch size variations. Uncertainty in the output manifests itself differently for different loss functions

EXPERIMENTS

Loss	Per-pixel acc.	Per-class acc.	Class IOU
L1	0.42	0.15	0.11
GAN	0.22	0.05	0.01
cGAN	0.57	0.22	0.16
L1+GAN	0.64	0.20	0.15
L1+cGAN	**0.66**	**0.23**	**0.17**
Ground truth	0.80	0.26	0.21

Table 1: FCN-scores for different losses, evaluated on Cityscapes labels↔photos.

Loss	Per-pixel acc.	Per-class acc.	Class IOU
Encoder-decoder (L1)	0.35	0.12	0.08
Encoder-decoder (L1+cGAN)	0.29	0.09	0.05
U-net (L1)	0.48	0.18	0.13
U-net (L1+cGAN)	**0.55**	**0.20**	**0.14**

Table 2: FCN-scores for different generator architectures (and objectives), evaluated on Cityscapes labels↔photos. (U-net (L1-cGAN) scores differ from those reported in other tables since batch size was 10 for this experiment and 1 for other tables, and random variation between training runs.)

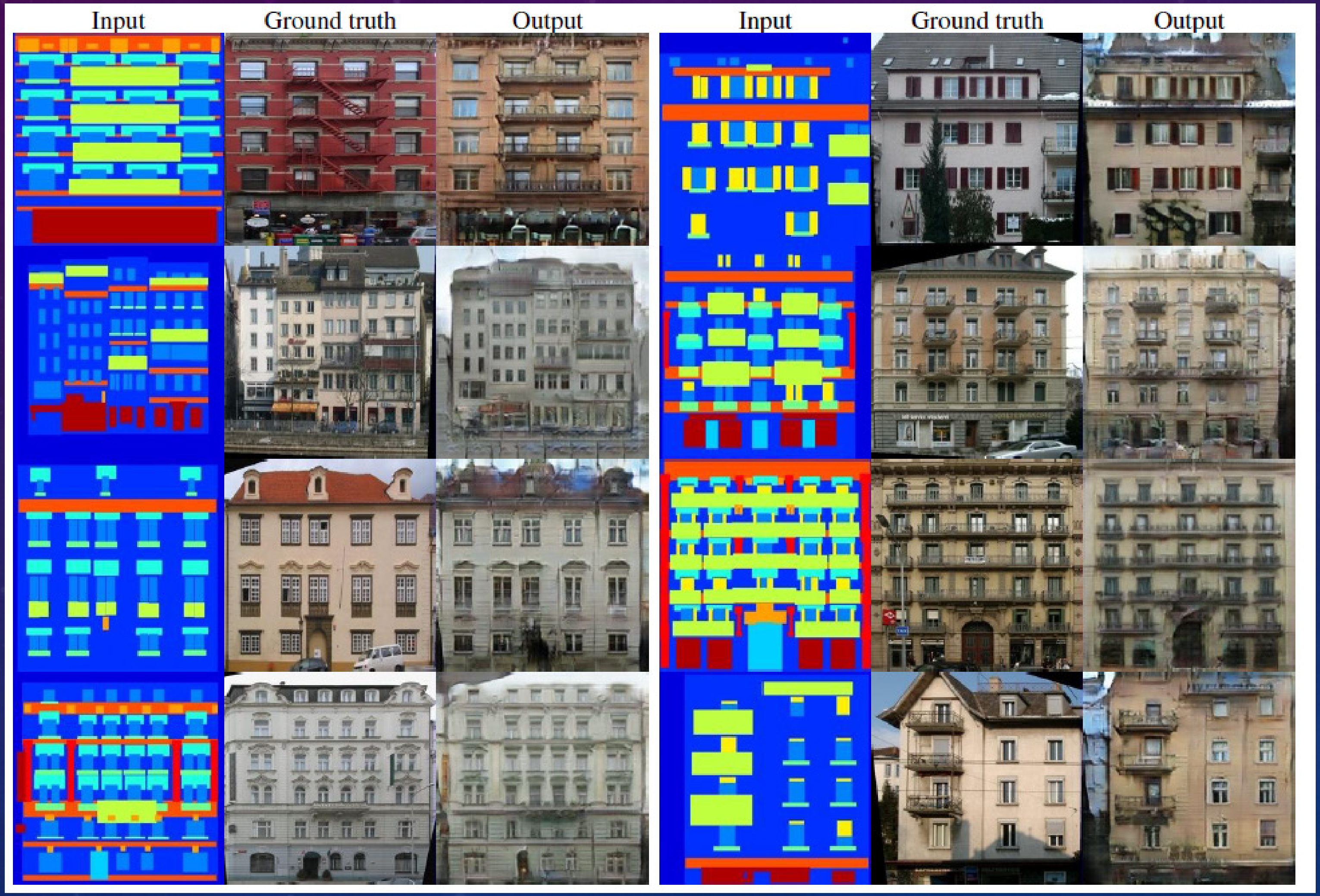
EXPERIMENTS

Discriminator receptive field	Per-pixel acc.	Per-class acc.	Class IOU
1×1	0.39	0.15	0.10
16×16	0.65	0.21	**0.17**
70×70	**0.66**	**0.23**	**0.17**
286×286	0.42	0.16	0.11

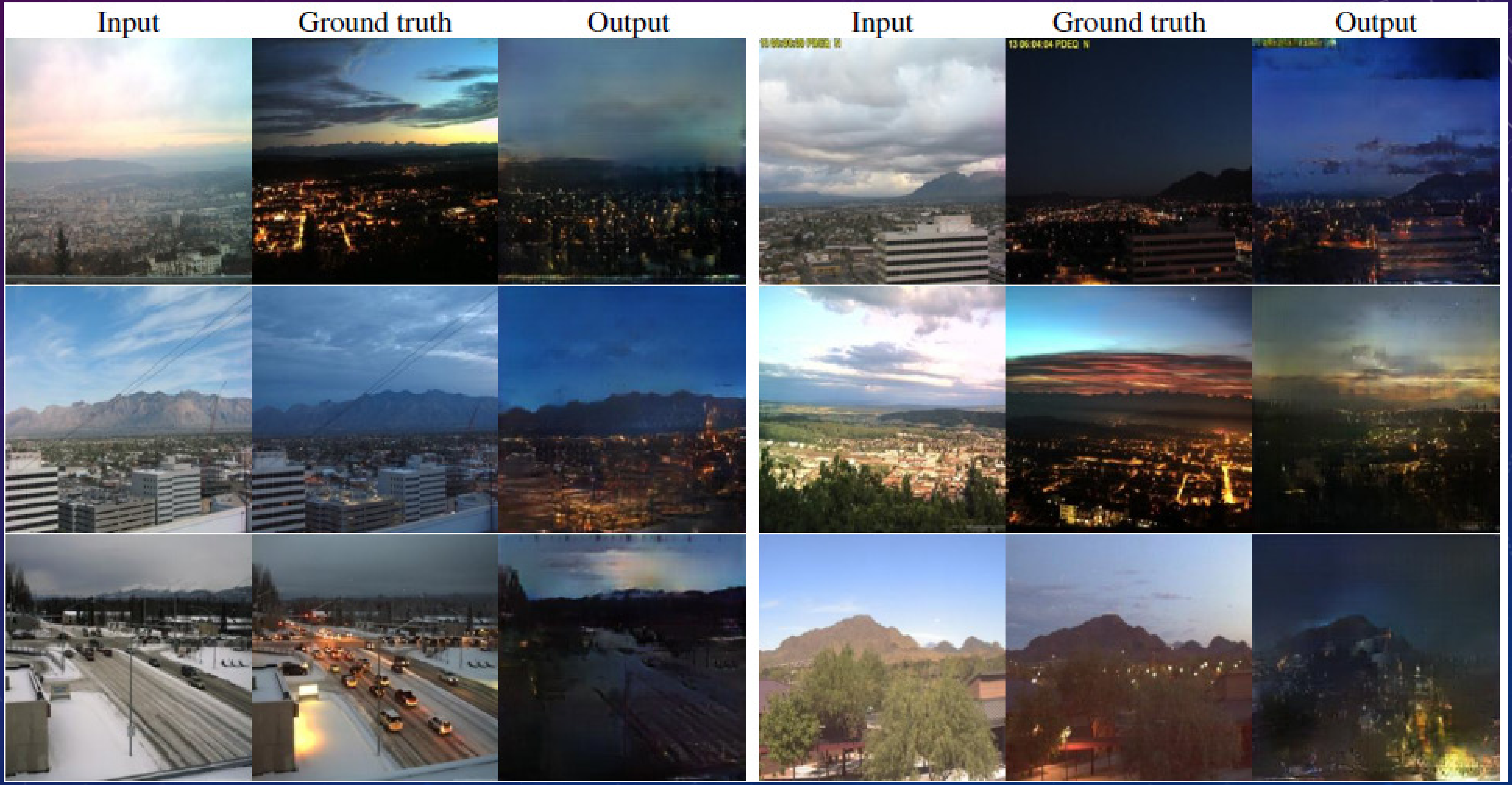
Table 3: FCN-scores for different receptive field sizes of the discriminator, evaluated on Cityscapes labels→photos. Note that input images are 256 × 256 pixels and larger receptive fields are padded with zeros.

EXPERIMENTS

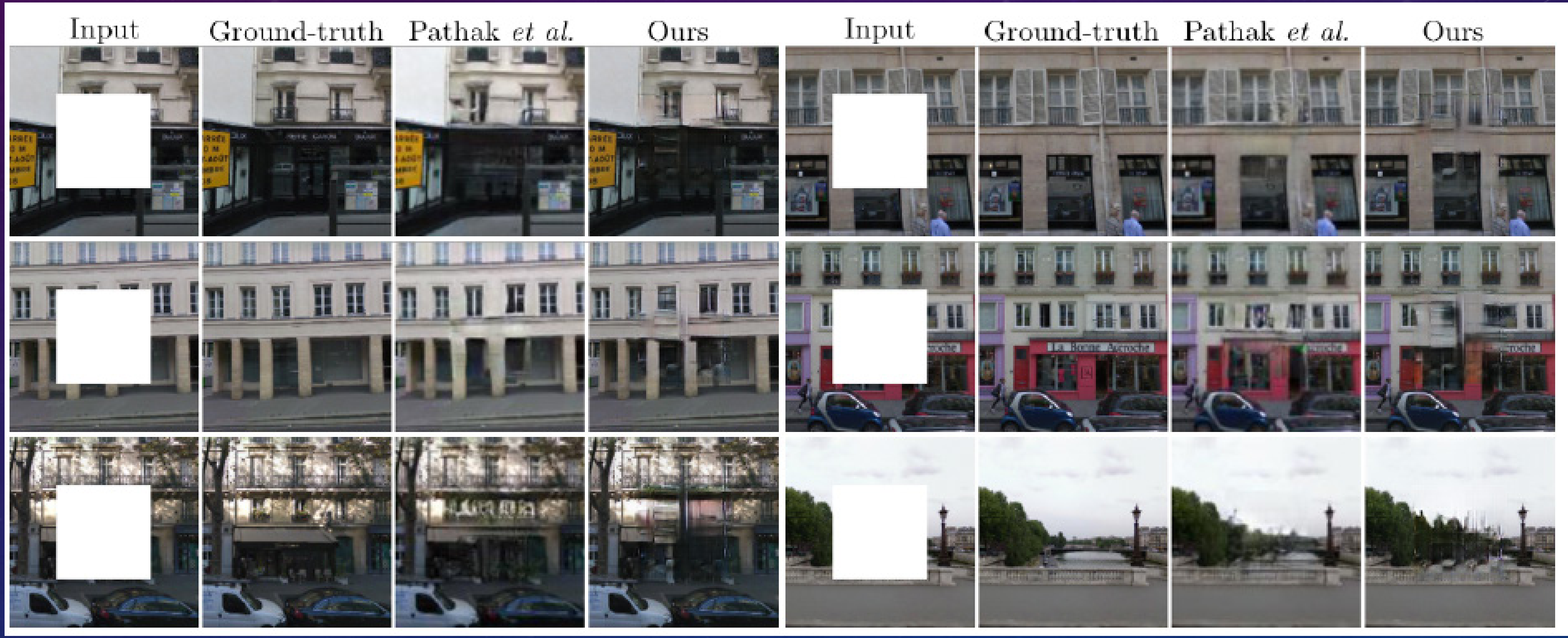
EXPERIMENTS



EXPERIMENTS



EXPERIMENTS



EXPERIMENTS

CONTENT

- Introduction
- Related work
- Method
- Experiments
- Conclusion

CONTENT

- Introduction
- Related work
- Method
- Experiments
- Conclusion

RESULTS

- The results in this paper suggest that conditional adver- sarial networks are a promising approach for many image- to-image translation tasks, especially those involving highly structured graphical outputs. These networks learn a loss adapted to the task and data at hand, which makes them applicable in a wide variety of settings.