23rd Vienna Deep Learning Meetup

31st January 2019 @ FH Technikum Wien

FH University of Applied Sciences TECHNIKUM WIEN

Vienna
Deep Learning
Meetup

The Organizers:

Alex Schindler
AIT & TU Wien

René Donner
contextflow

Thomas Lidy
Musimap

Jan Schlüter
OFAI & UTLN

Vienna Deep Learning Meetup

Agenda:

- Welcome
- Explainable Neural Symbolic Learning (by Ahmad Haj Mosa and Fabian Schneider, PwC)
- <break 15 min>
- Interesting Papers & Trends: NeurIPS 2018 (by Rene Donner, Contextflow)
- Networking and Discussions

Vienna Deep Learning Meetup

Announcements

VDLM on Github

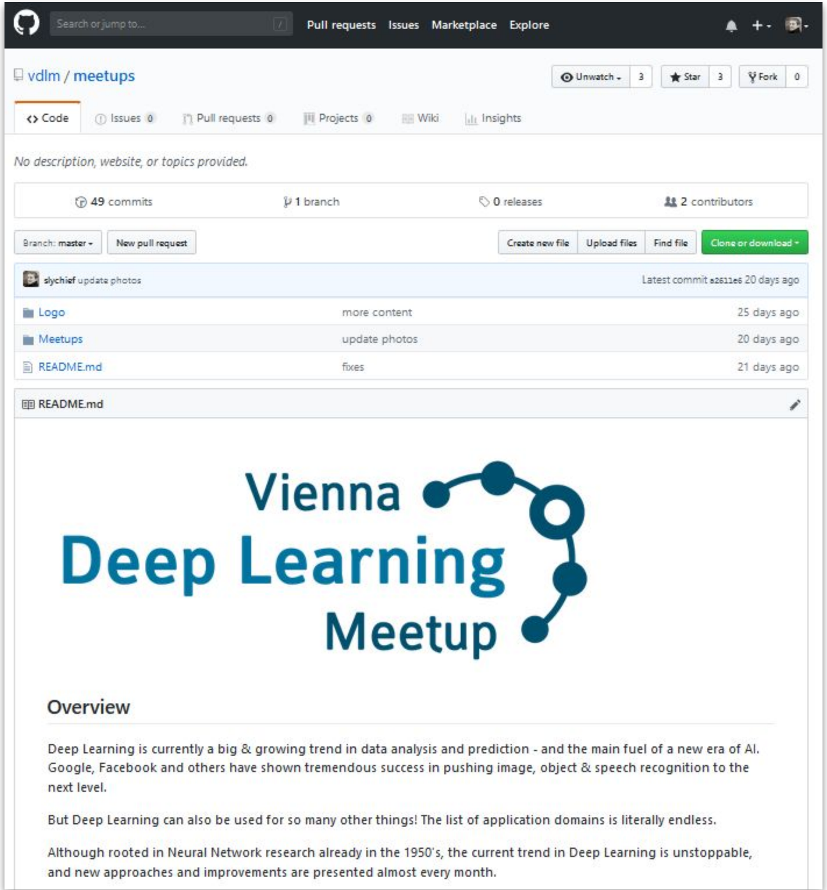
- ❑ history
- ❑ slides
- ❑ videos
- ❑ Wiki

Meetups

#	Date	Place	Topic	Link	Video	Meetup.com
1	2016-04-07	Sector 5	intro	more		link
2	2016-05-09	Sector 5		more		link
3	2016-06-06	Sector 5		more		link
4	2016-07-07	TU Wien		more		link
5	2016-09-22	Automic Software GmbH		more		link
6	2016-10-12	Sector 5		more		link
7	2016-12-01	Agentur Virtual Identity		more		link
8	2017-01-17	TU Wien Informatik		more		link
9	2017-02-21	bwin.party services (Austria) GmbH		more		link

Talks

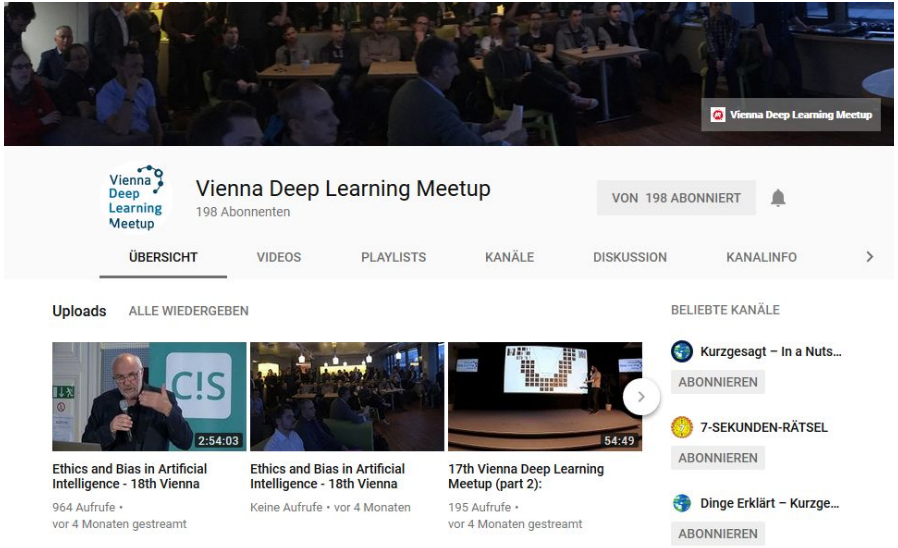
Date	MU#	Speaker	Topic	Slides
2016-04-07	1	Thomas Lidy	An overview presentation of Deep Learning	pdf
2016-04-07	1	Jan Schlüter	History, Approaches, Applications	pdf
2016-05-09	2	Alex Champandard	Neural Networks for Image Synthesis	
2016-05-09	2	Gregor Mitscha-Baude	Recurrent Neural Networks	pdf
2016-06-06	3	Jan Schlüter	Open-source Deep Learning with Theano and Lasagne	pdf
2016-09-22	5	Josef Puchinger	Deep Learning & The Future of Automation	
2016-09-22	5	Christoph Körner	Going Deeper with GoogLeNet and CaffeJS	pdf



https://github.com/vdlm/meetups

Vienna Deep Learning Meetup

VDLM Youtube Channel



https://www.youtube.com/ViennaDeepLearningMeetup

Vienna Deep Learning Meetup

keep current

Applied Deep Learning course

based on

fast.ai "Practical Deep Learning for Coders" (v3) MOOC

PyTorch 1.0

Contact: Liad Magen

https://keepcurrent.online/ml-course.html

Machine Learning Prague 2019

FEBRUARY 22 - 24

1000+
ATTENDEES

45
SPEAKERS

8
WORKSHOPS

Code 20% off - vdlm20


Machine Learning Prague 2019
FEBRUARY 22 - 24
1000+
ATTENDEES
45
SPEAKERS
8
WORKSHOPS
Raffle Winner: Markus Pak


Machine Learning Prague 2019
FEBRUARY 22 - 24
1000+
ATTENDEES
45
SPEAKERS
8
WORKSHOPS
Code 20% off - vdlm20
New Raffle: Win 1 Ticket


Machine Learning Prague 2019
FEBRUARY 22 - 24
1000+
ATTENDEES
45
SPEAKERS
8
WORKSHOPS
Code 20% off - vdlm20
Winner of Today: Michael Pieler

Hot Topics &
Latest News

a short block at every meetup
to briefly present recent papers and news

Send us contributions (tom.lidy@gmail.com)
or come with slides to do a short block yourself!

Vienna Deep Learning Meetup

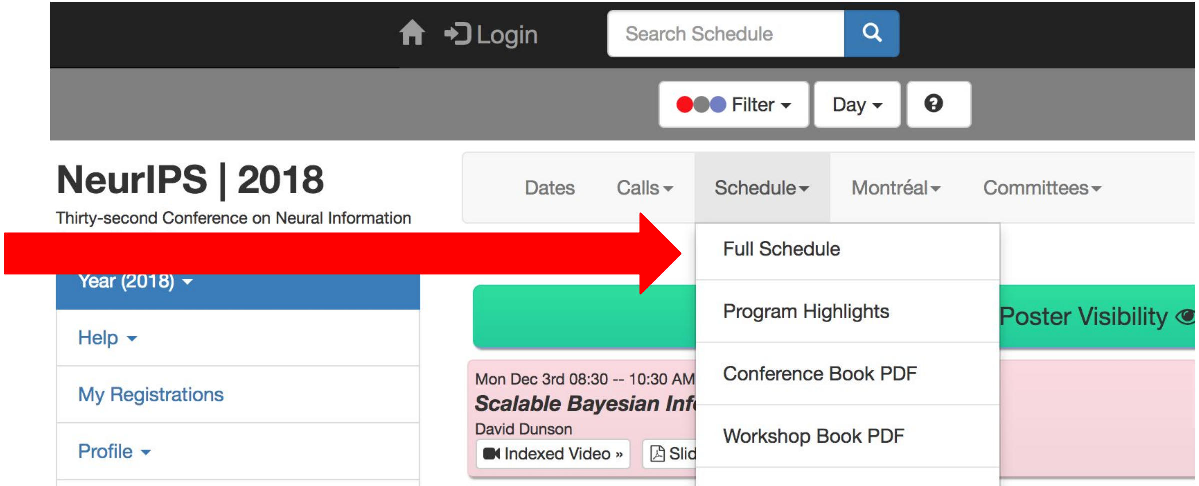
Interesting Papers & Trends
NeurIPS 2018

Interesting Papers & Trends NeurIPS 2018

- **Network architectures**
  - Perturbative NNs
  - Loss Function Visualization
  - Neural Differential Equations
- **GANs**
  - Vid2vid Video Generation
  - Text influenced Image Generation
- **Assorted Papers**
  - Dropblock Regularization
  - Features-Replay Parallel Training
  - DIfNet Semantic segmenation
- **Industry News**
  - Graphotate
  - Graphcore
  - News from NVidia

Interesting Papers & Trends – NeurIPS 2018

- **Largest DL/AI conference in the world**
- **Sold out in 11min**
- **All papers online: https://nips.cc/Conferences/2018/Schedule**
- **Papers with code: https://paperswithcode.com/conference/nips-2018**



Interesting Papers & Trends NeurIPS 2018

- **Network architectures**
  - Perturbative NNs (CVPR 18 ;-)
  - Loss Function Visualization
  - Neural Differential Equations
- **GANs**
  - Vid2vid Video Generation
  - Text influenced Image Generation
- **Assorted Papers**
  - Dropblock Regularization
  - Features-Replay Parallel Training
  - DIfNet Semantic segmenation
- **Industry News**
  - Graphotate
  - Graphcore
  - News from NVidia
  - AI Report 2018

Perturbative Neural Networks

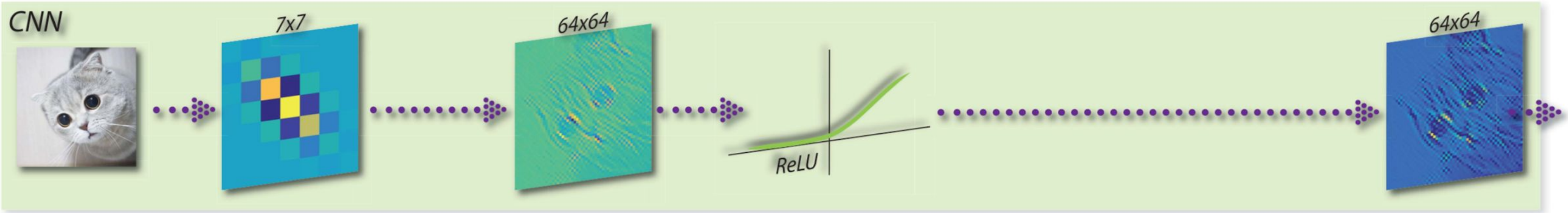
Felix Juefei-Xu
Carnegie Mellon University
felixu@cmu.edu

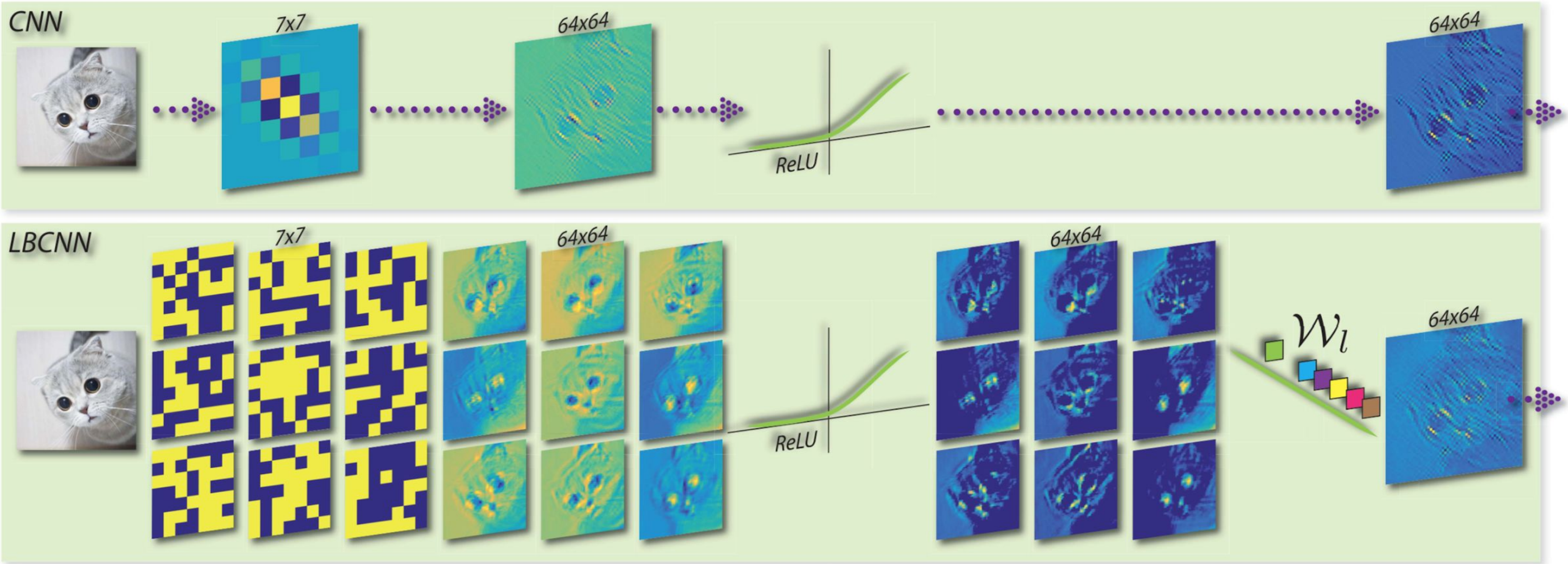
Vishnu Naresh Boddeti
Michigan State University
vishnu@msu.edu

Marios Savvides
Carnegie Mellon University
msavvid@ri.cmu.edu

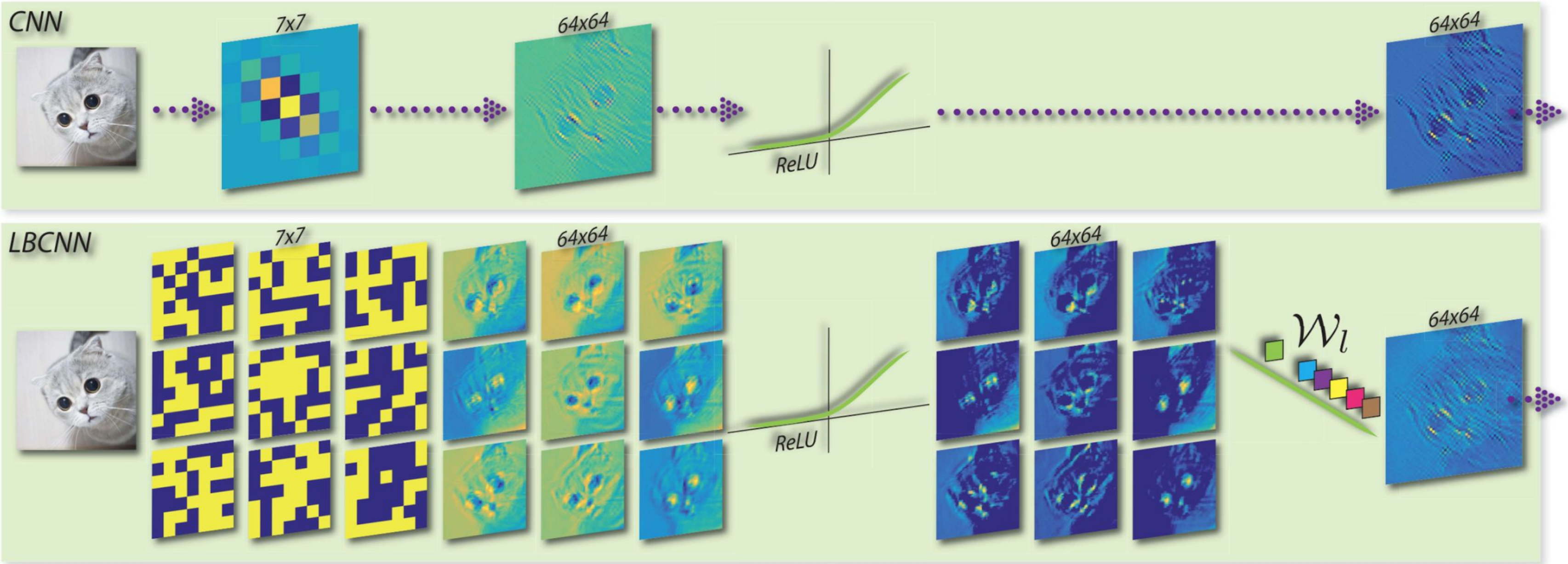
Images - CNNs
Text / Time Series - RNNs

Right?

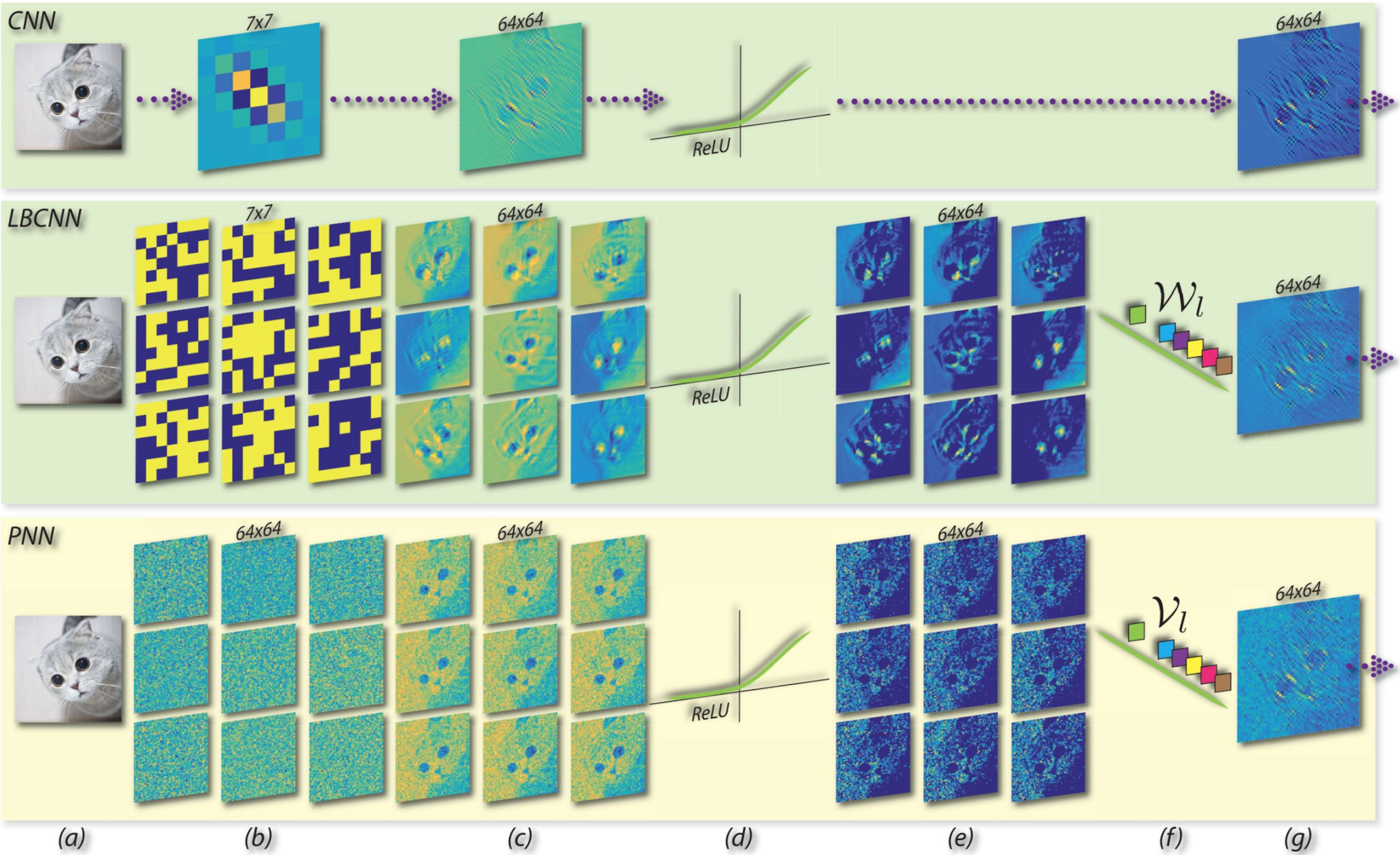

CNN
7x7
64x64
ReLU
64x64



CNN
7x7
64x64
ReLU
64x64
LBCNN
7x7
64x64
ReLU
64x64
$\mathcal{W}_l$
64x64



CNN
7x7
64x64
ReLU
64x64
LBCNN
7x7
64x64
ReLU
64x64
$\mathcal{W}_l$
64x64


CNN
7x7
64x64
ReLU
64x64
LBCNN
7x7
64x64
ReLU
64x64
$\mathcal{W}_l$
64x64
PNN
64x64
64x64
ReLU
64x64
$\mathcal{V}_l$
64x64
(a)
(b)
(c)
(d)
(e)
(f)
(g)

$$\frac{\text{\# param. in CNN}}{\text{\# param. in PNN}} = \frac{p \cdot h \cdot w \cdot q}{m \cdot q} = \frac{p \cdot h \cdot w}{m}$$

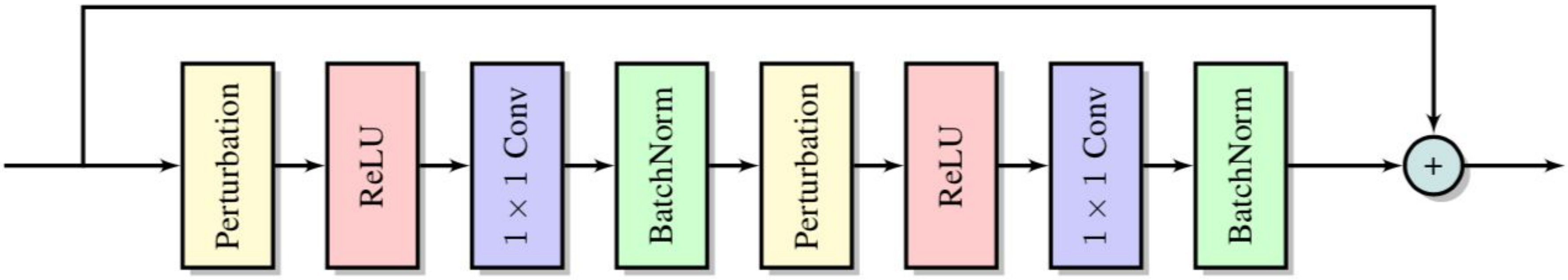


Figure 4: Perturbation residual module.

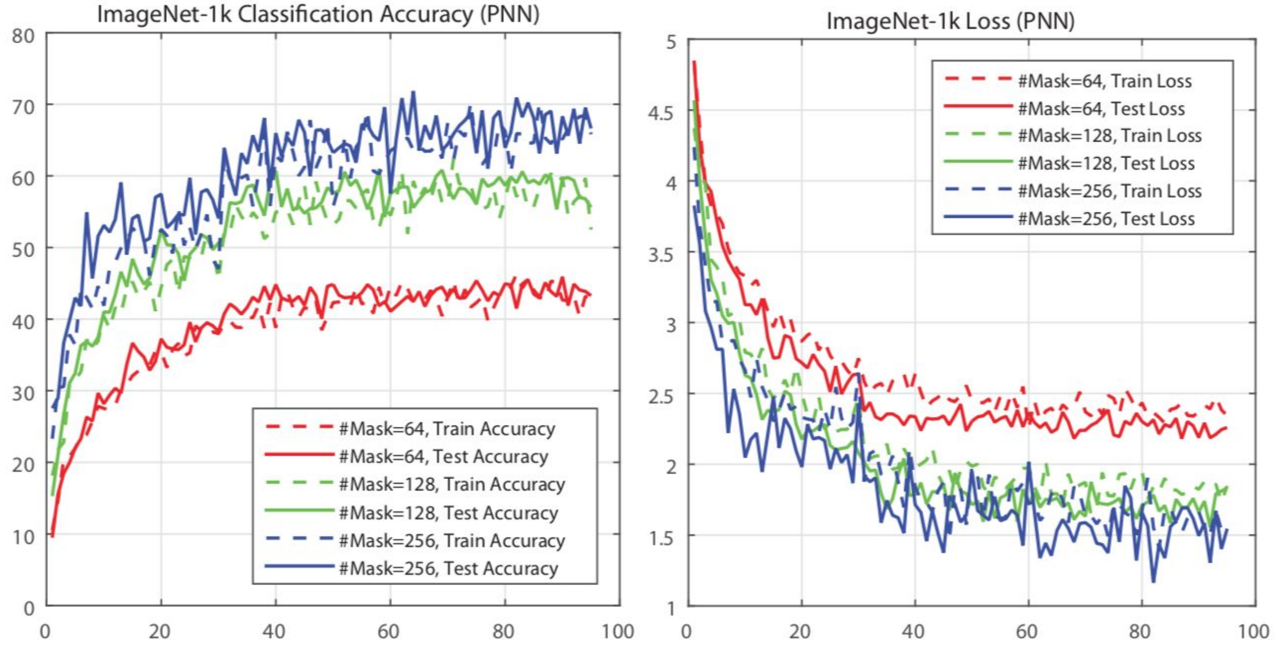


Figure 5: Accuracy and loss on ImageNet-1k classification using PNN (ResNet-18) with various number of perturbation masks per layer.

Table 2: Classification accuracy (%) on ImageNet-1k (PNN vs. CNN)

PNN-ResNet-18	ResNet-18	**PNN-ResNet-50**	ResNet-50
71.84	69.57	76.23	75.99

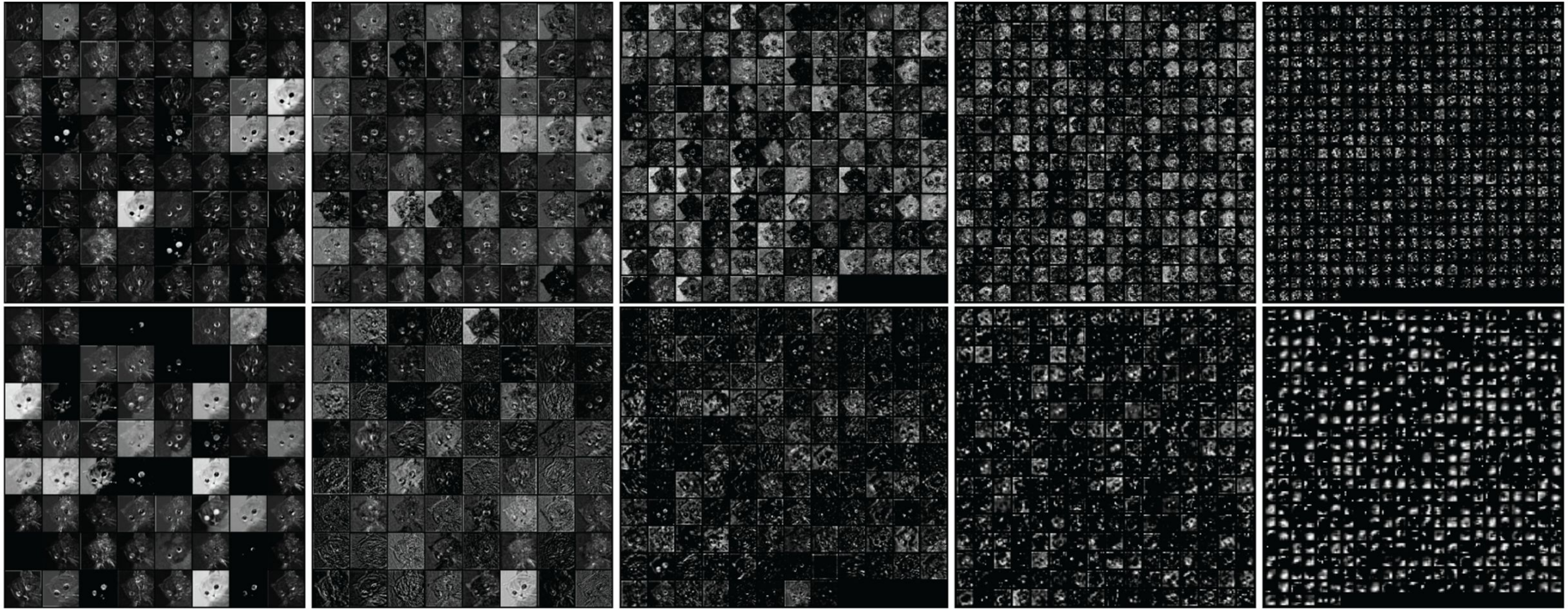
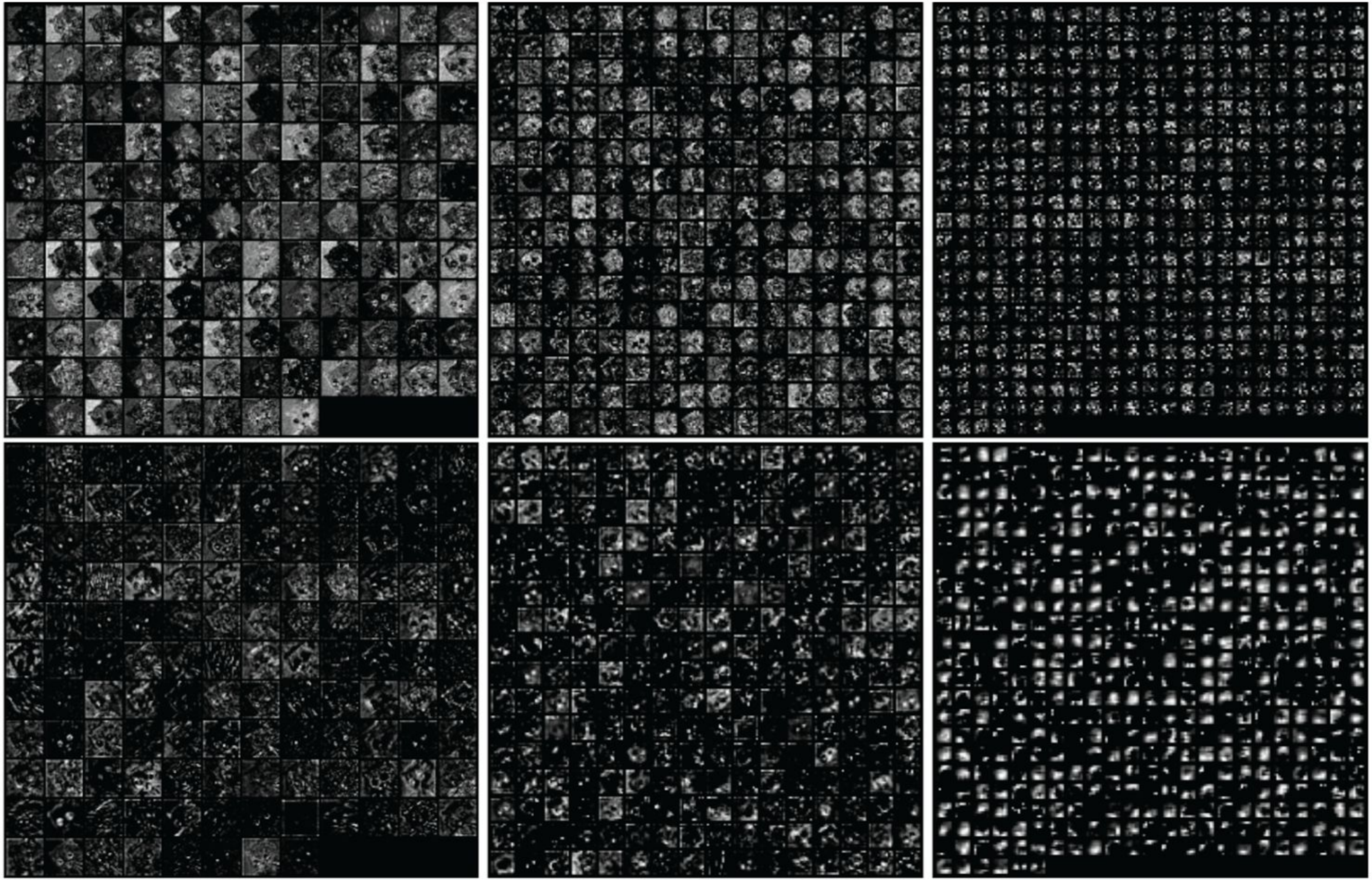
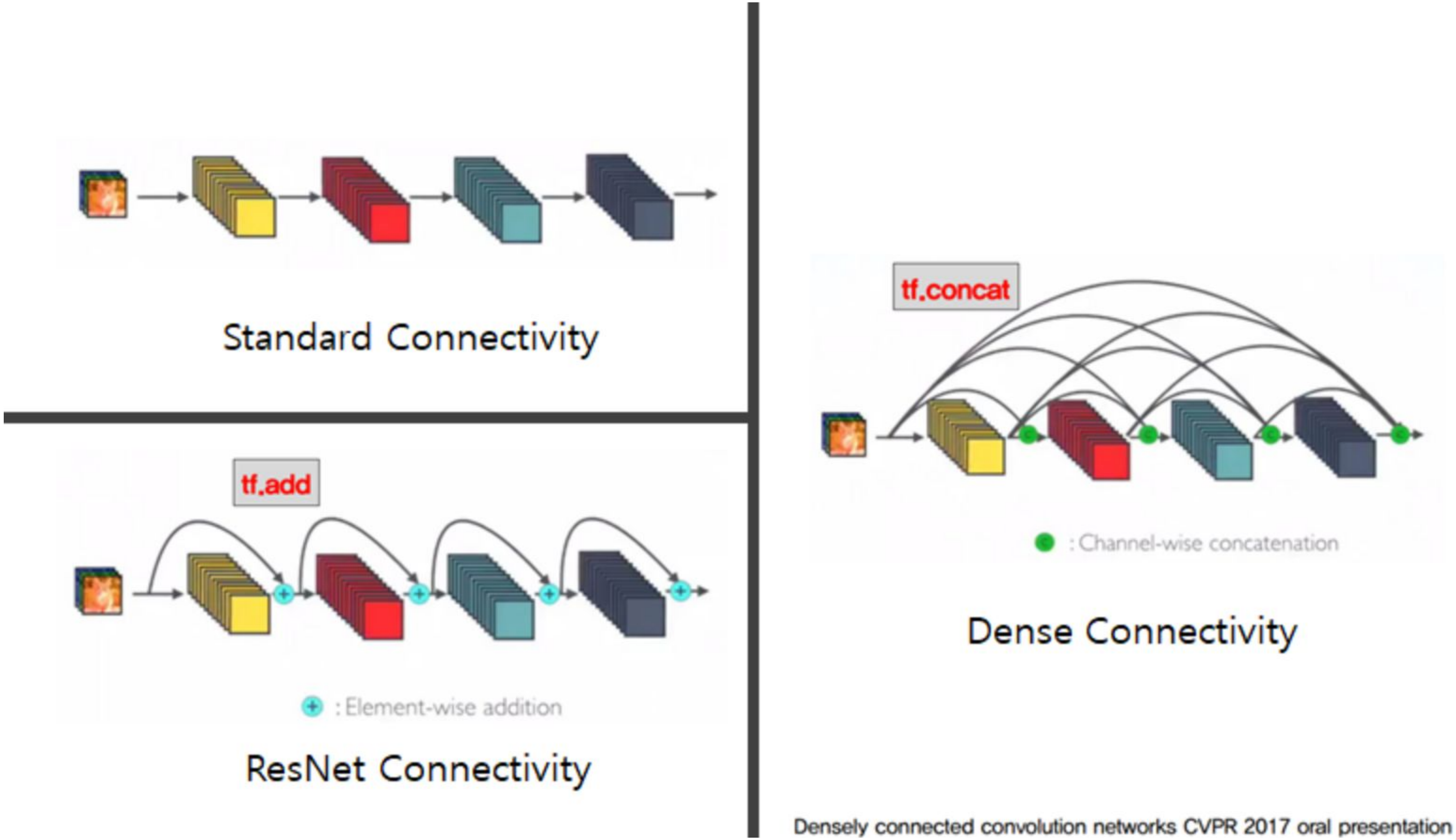


Figure 3: Feature maps of the same image from different layers of 18-layer ResNet architecture trained on the ImageNet dataset. The top row corresponds to PNN, while the bottom row corresponds to CNN.



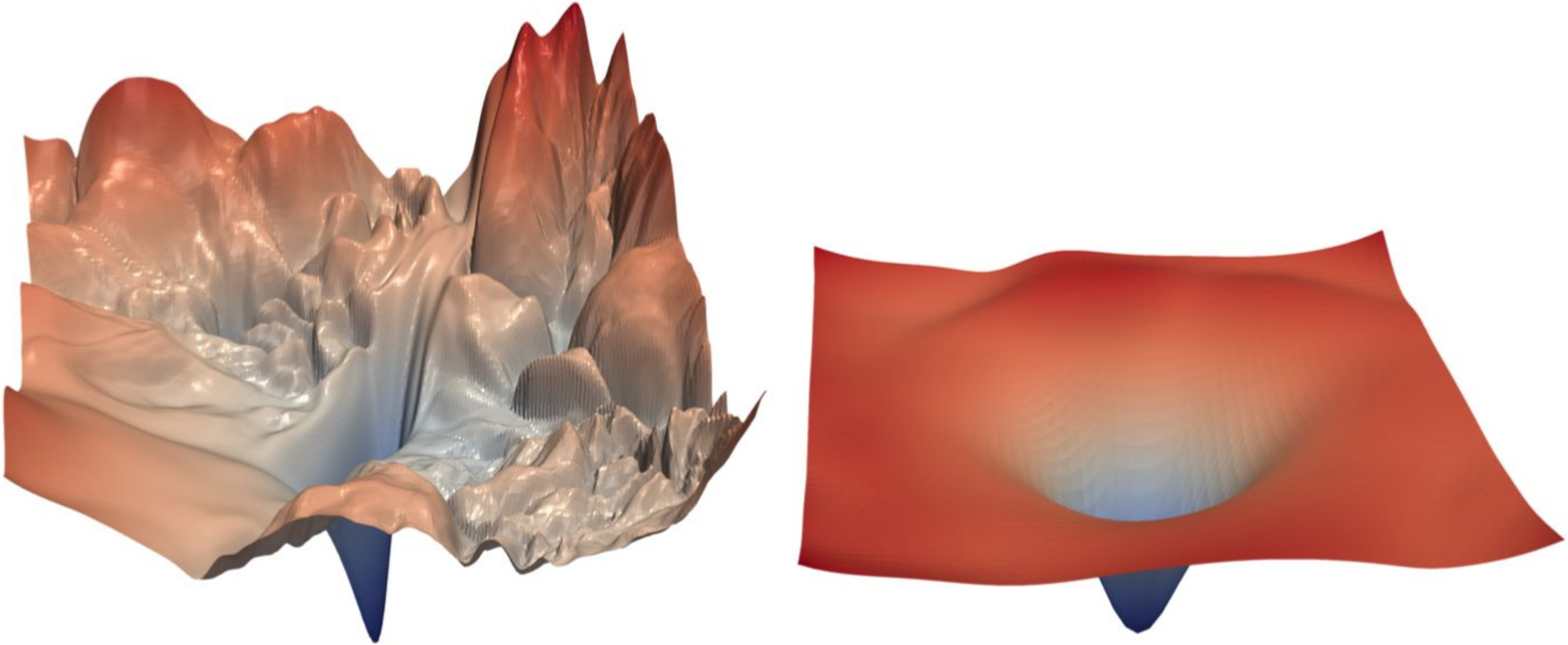
Visualizing the Loss Landscape of Neural Nets

Hao Li[1], **Zheng Xu**[1], **Gavin Taylor**[2], **Christoph Studer**[3], **Tom Goldstein**[1]
[1]University of Maryland, College Park [2]United States Naval Academy [3]Cornell University
{haoli,xuzh,tomg}@cs.umd.edu, taylor@usna.edu, studer@cornell.edu



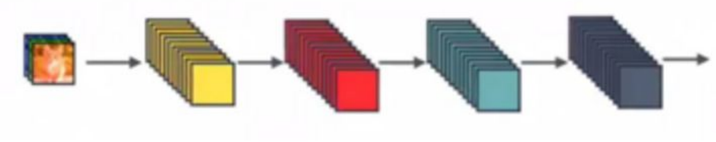
Densely connected convolution networks CVPR 2017 oral presentation slide

https://chatbotslife.com/resnets-highwaynets-and-densenets-oh-my-9bb15918ee32

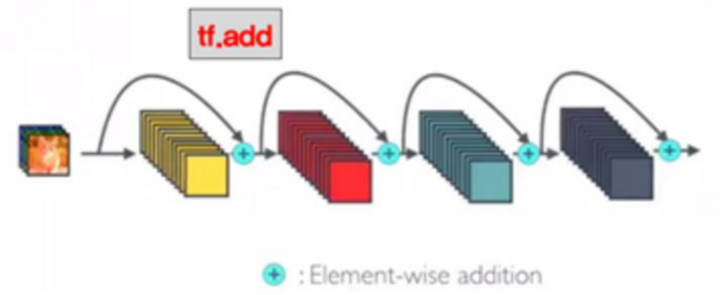


(a) without skip connections (b) with skip connections

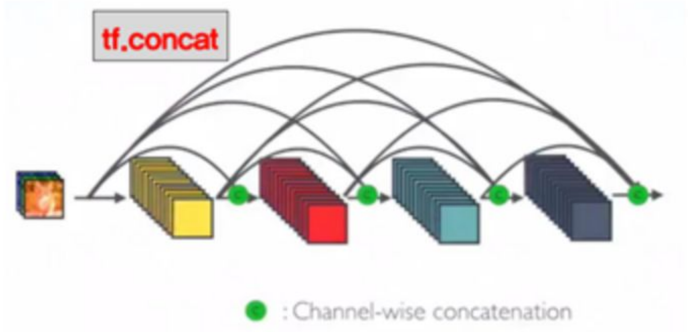
Figure 1: The loss surfaces of ResNet-56 with/without skip connections. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures.



Standard Connectivity



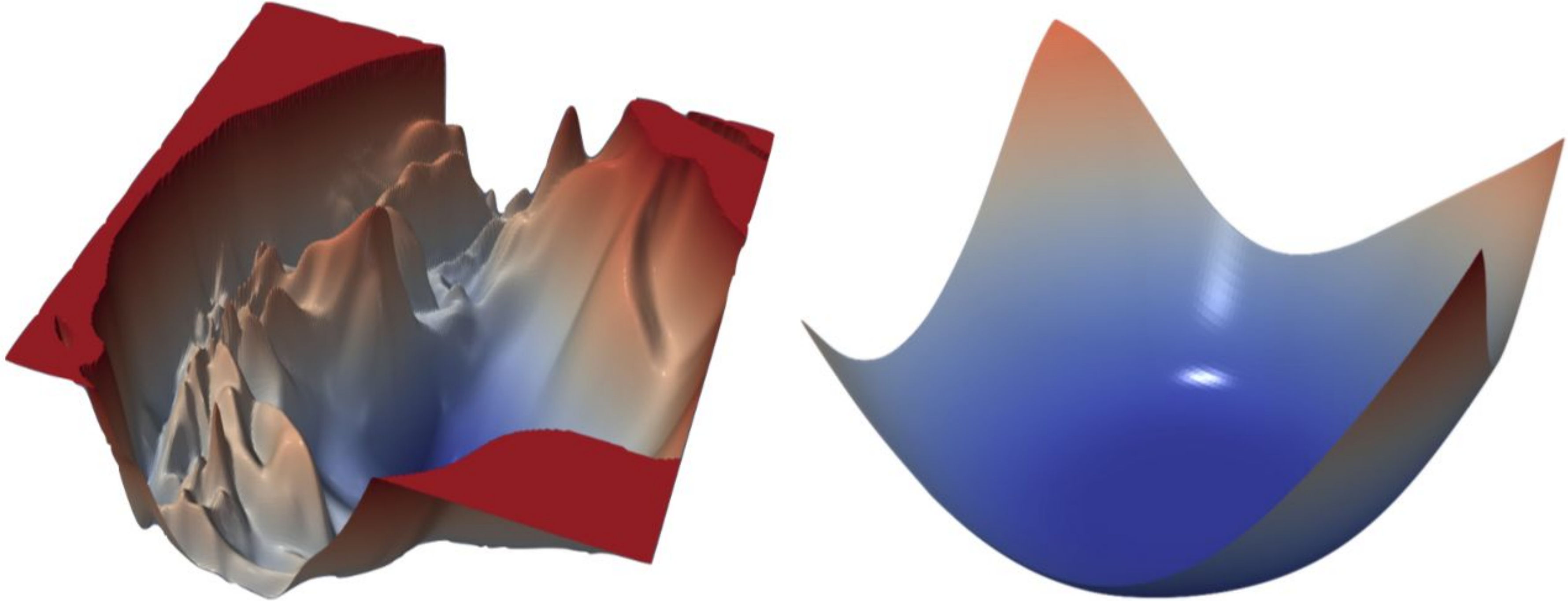
ResNet Connectivity



Dense Connectivity

Densely connected convolution networks CVPR 2017 oral presentation slide

https://chatbotslife.com/resnets-highwaynets-and-densenets-oh-my-9bb15918ee32



(a) ResNet-110, no skip connections

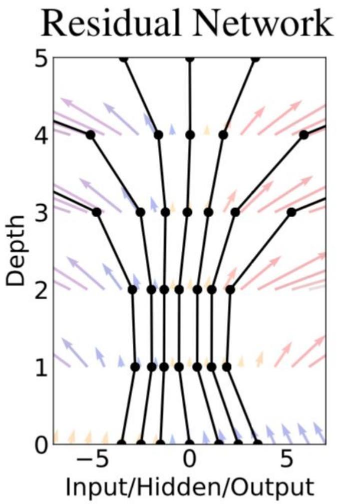
(b) DenseNet, 121 layers

Figure 4: The loss surfaces of ResNet-110-noshort and DenseNet for CIFAR-10.

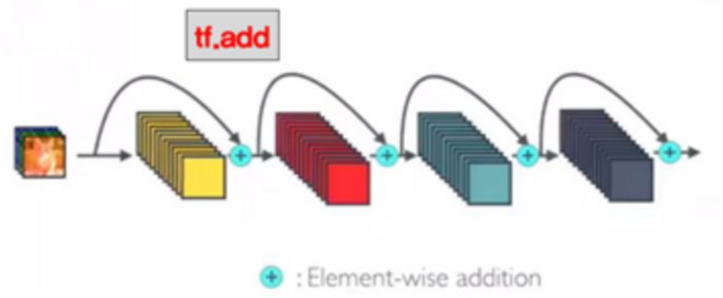
Neural Ordinary Differential Equations

Ricky T. Q. Chen*, Yulia Rubanova*, Jesse Bettencourt*, David Duvenaud
University of Toronto, Vector Institute
{rtqichen, rubanova, jessebett, duvenaud}@cs.toronto.edu

Resnet vs ODE



$$\mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, \theta_t)$$

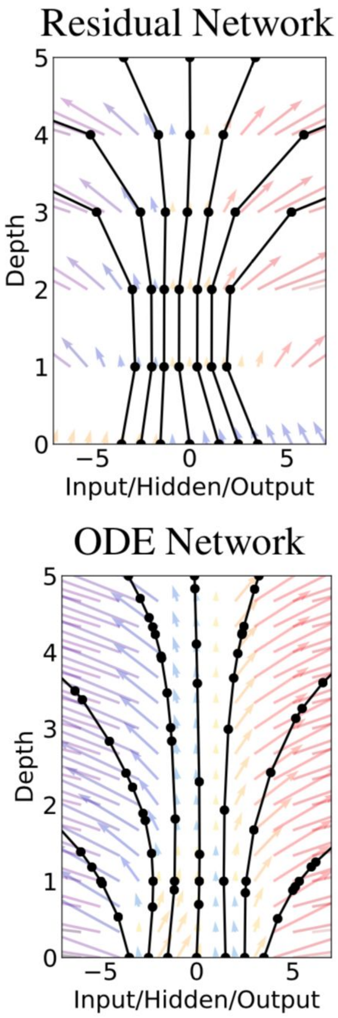


ResNet Connectivity

Resnet vs ODE

$$\mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, \theta_t)$$

$$\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta)$$



Advantages

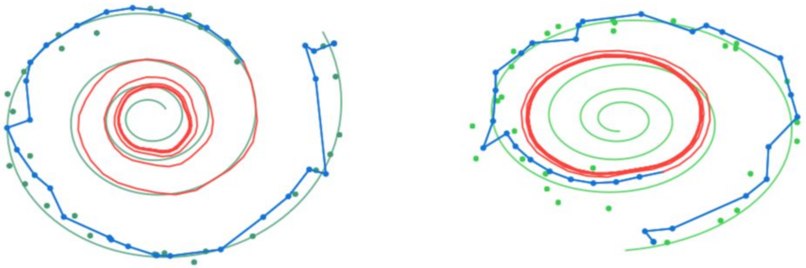
- **Memory efficiency**
  - “[compute gradients] without backpropagating through the operations of the solver. Not storing any intermediate quantities of the forward pass allows us to train our models with constant memory cost as a function of depth, a major bottleneck of training deep model”.
- **Adaptive computation**
  - Modern ODE solvers provide guarantees about the growth of approximation error, monitor the level of error & adapt their evaluation strategy on the fly to achieve the requested level of accuracy
  - After training, accuracy can be reduced for real-time or low-power applications.

Table 1: Performance on MNIST. †From LeCun et al. (1998).

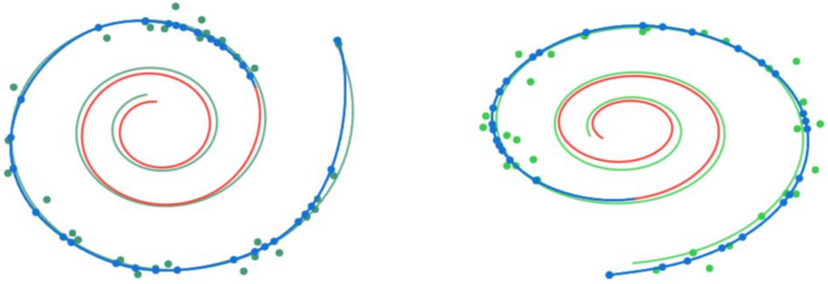
	Test Error	# Params	Memory	Time
1-Layer MLP†	1.60%	0.24 M	-	-
ResNet	0.41%	0.60 M	$\mathcal{O}(L)$	$\mathcal{O}(L)$
RK-Net	0.47%	0.22 M	$\mathcal{O}(\tilde{L})$	$\mathcal{O}(\tilde{L})$
ODE-Net	0.42%	0.22 M	$\boldsymbol{\mathcal{O}(1)}$	$\mathcal{O}(\tilde{L})$

A generative latent function time-series model

- Applying neural networks to irregularly-sampled data such as medical records, network traffic, or neural spiking data is difficult.
- -> Continuous-time, generative approach to modeling time series



(a) Recurrent Neural Network



(b) Latent Neural Ordinary Differential Equation

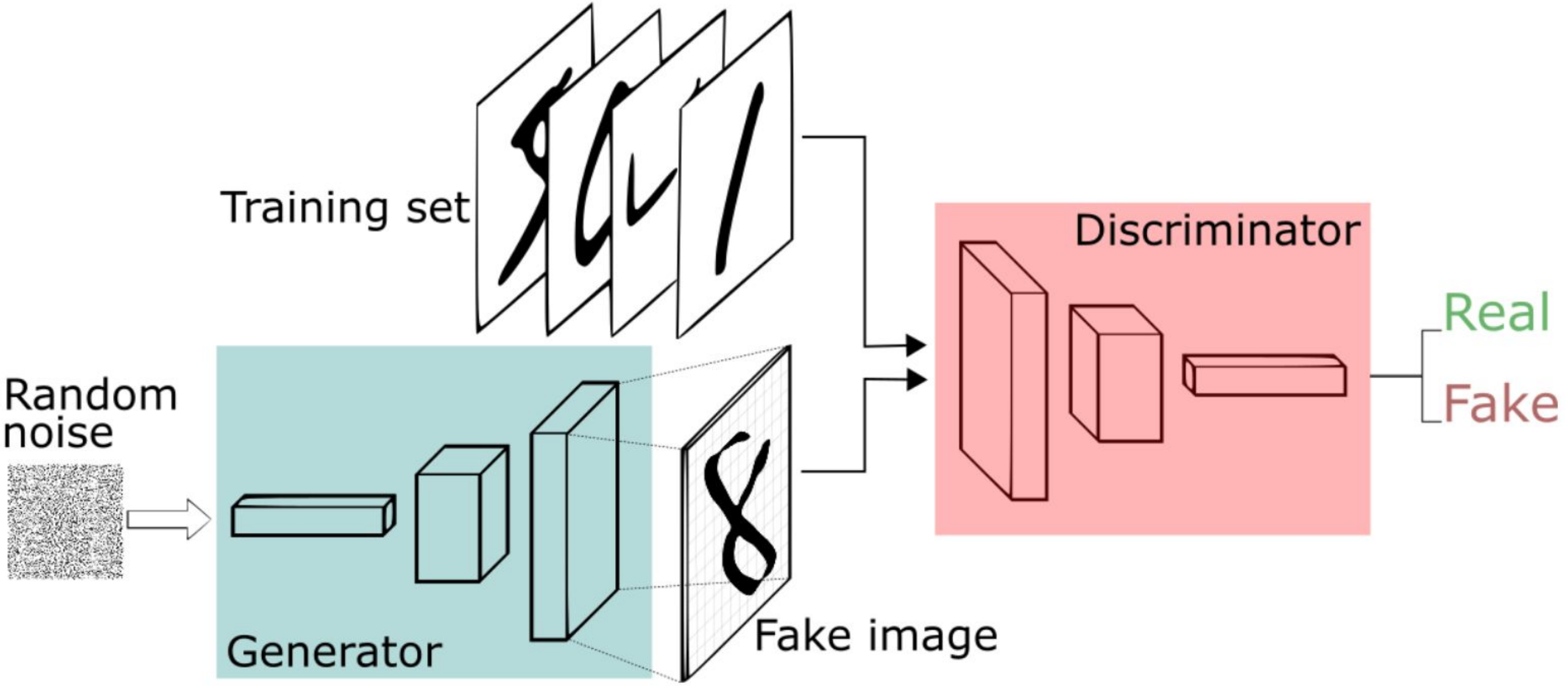
Contribution

- **Black-box ODE solvers as a model component**
- **Allow an ODE solver to be trained end-to-end with any other differentiable model components!**
- **New models for time-series modeling, supervised learning, and density estimation**
- **https://github.com/rtqichen/torchdiffeq**

Interesting Papers & Trends NeurIPS 2018

- **Network architectures**
  - Perturbative NNs (CVPR 18 ;-)
  - Loss Function Visualization
  - Neural Differential Equations
- **GANs**
  - Vid2vid Video Generation
  - Text influenced Image Generation
- **Assorted Papers**
  - Dropblock Regularization
  - Features-Replay Parallel Training
  - DIfNet Semantic segmenation
- **Industry News**
  - Graphotate
  - Graphcore
  - News from NVidia
  - AI Report 2018

GANS - Generative Adverserial Networks



https://sthalles.github.io/intro-to-gans/

Text-Adaptive Generative Adversarial Networks: Manipulating Images with Natural Language

Seonghyeon Nam, Yunji Kim, and Seon Joo Kim
Yonsei University
{shnnam,kim_yunji,seonjookim}@yonsei.ac.kr

This particular bird with a **red head and breast** and features **grey wings**.

This small bird has a **blue crown** and **white belly**.

Original [11] [15] Ours

Figure 1: Examples of image manipulation using natural language description. Existing methods produce reasonable results, but fail to preserve text-irrelevant contents such as the background of the original image. In comparison, our method accurately manipulates images according to the text while preserving text-irrelevant contents.

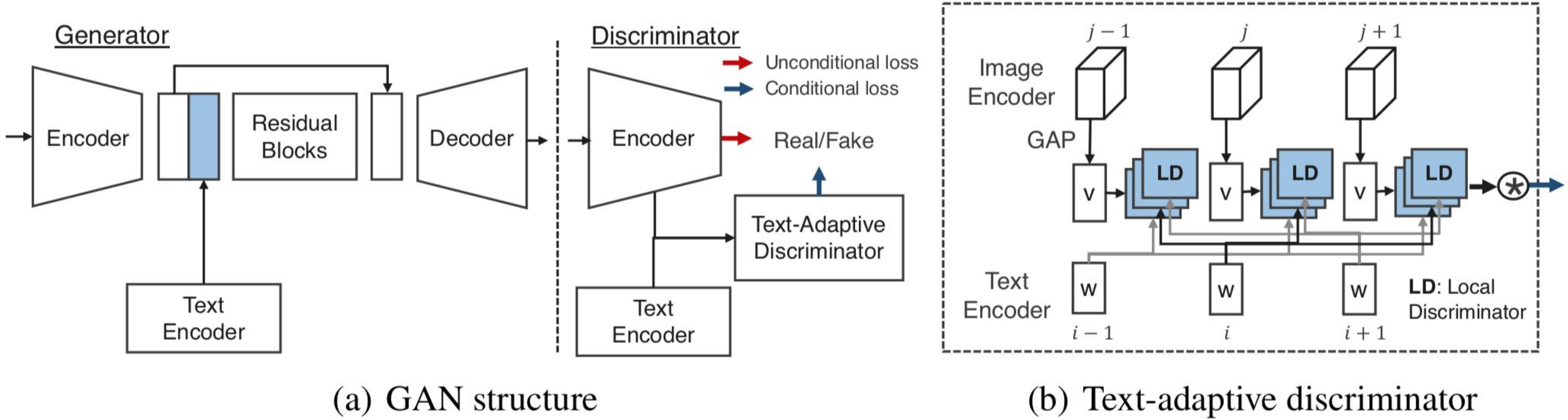


Figure 2: The proposed GAN structure. (a) shows the overall GAN architecture and (b) depicts our text-adaptive discriminator. In (b), the attention and the layer-wise weight are omitted for simplicity.

Original

This bird has **wings that are blue** and has a **white belly**.

A small bird with **white base** and **black stripes** throughout its belly, head, and feathers.

Original

The petals of the flower have **yellow and red stripes**.

This flower has petals of **pink and white color** with **yellow stamens**.

This is a **black bird** with **gray and white wings** and a **bright yellow belly and chest**.

This flower has **petals that are white** and has **patches of yellow**.

This **pink flower** has **long and oval petals** and a **large yellow stamen**.

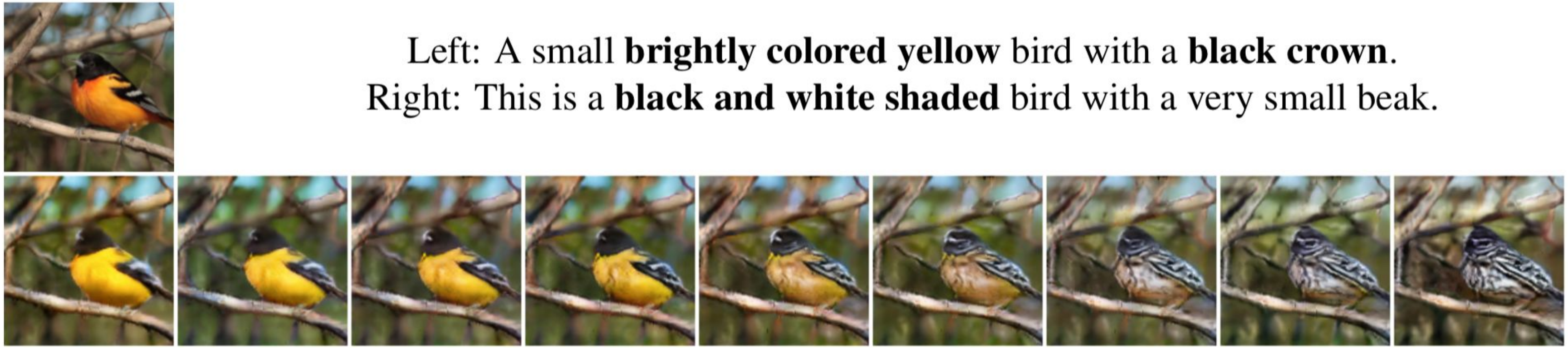


Figure 7: Sentence interpolation results. Our generator smoothly generates new visual attributes without loosing original image.

Video-to-Video Synthesis

Ting-Chun Wang[1], **Ming-Yu Liu**[1], **Jun-Yan Zhu**[2], **Guilin Liu**[1],
Andrew Tao[1], **Jan Kautz**[1], **Bryan Catanzaro**[1]
[1]NVIDIA, [2]MIT CSAIL
{tingchunw,mingyul,guilinl,atao,jkautz,bcatanzaro}@nvidia.com,
junyanz@mit.edu

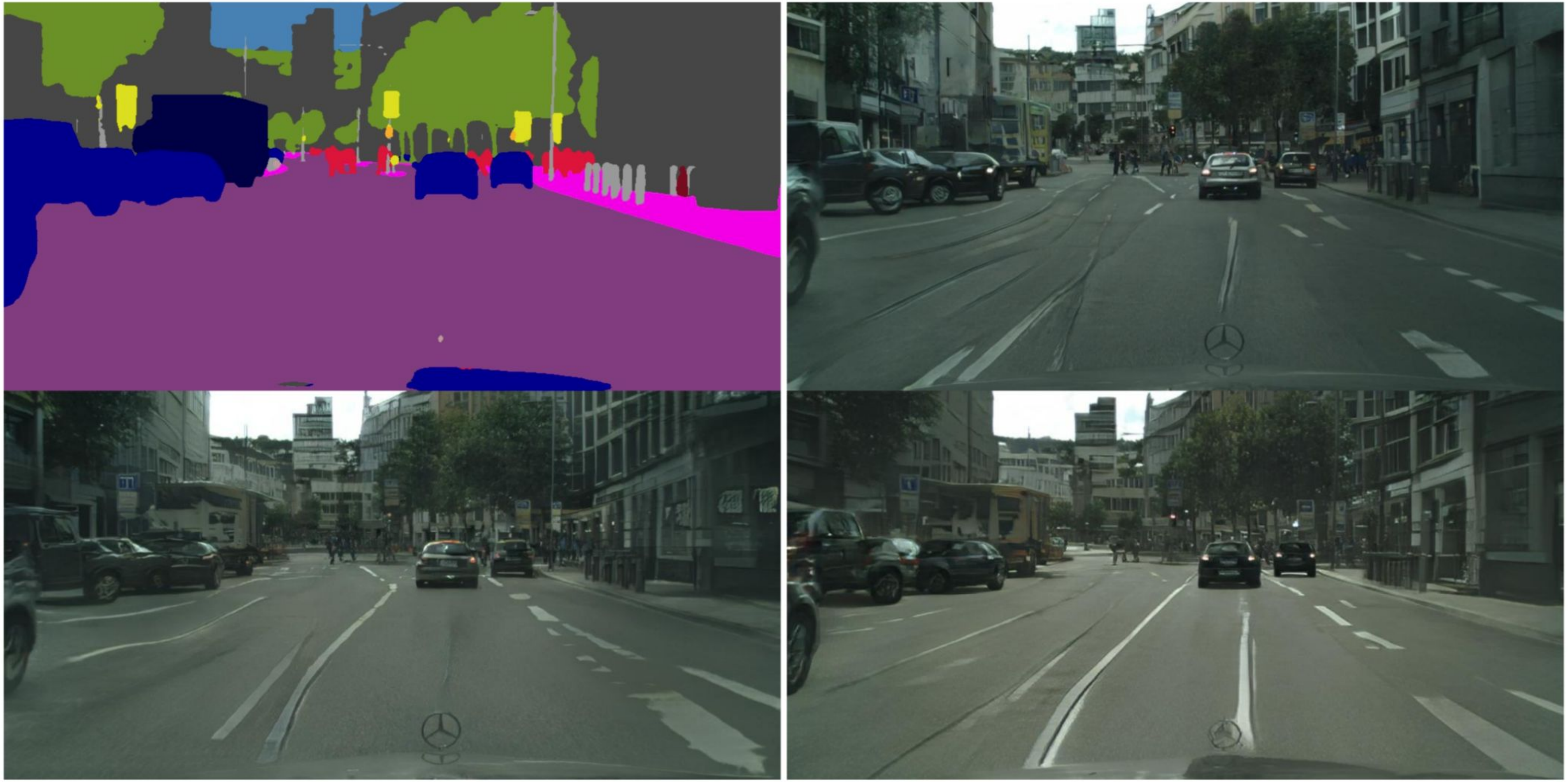


Figure 1: Generating a photorealistic video from an input segmentation map video on Cityscapes. Top left: input. Top right: `pix2pixHD`. Bottom left: `COVST`. Bottom right: `vid2vid` (ours). *Click the image to play the video clip in a browser.*

https://tcwang0509.github.io/vid2vid/paper_gifs/cityscapes_comparison.gif

vid2vid related work

- **Generative Adversarial Networks (GANs)**
- **Image-to-image translation**
- **Unconditional video synthesis**
- **Future video prediction**

Figure 3: Example multi-modal video synthesis results. These synthesized videos contain different road surfaces. *Click the image to play the video clip in a browser.*

https://tcwang0509.github.io/vid2vid/paper_gifs/cityscapes_change_styles.gif

Figure 4: Example results of changing input semantic segmentation masks to generate diverse videos. Left: tree→building. Right: building→tree. The original video is shown in Figure 3. *Click the image to play the video clip in a browser.*

https://tcwang0509.github.io/vid2vid/paper_gifs/cityscapes_change_labels.gif

Figure 5: Example face→sketch→face results. Each set shows the original video, the extracted edges, and our synthesized video. *Click the image to play the video clip in a browser.*

https://tcwang0509.github.io/vid2vid/paper_gifs/face.mp4

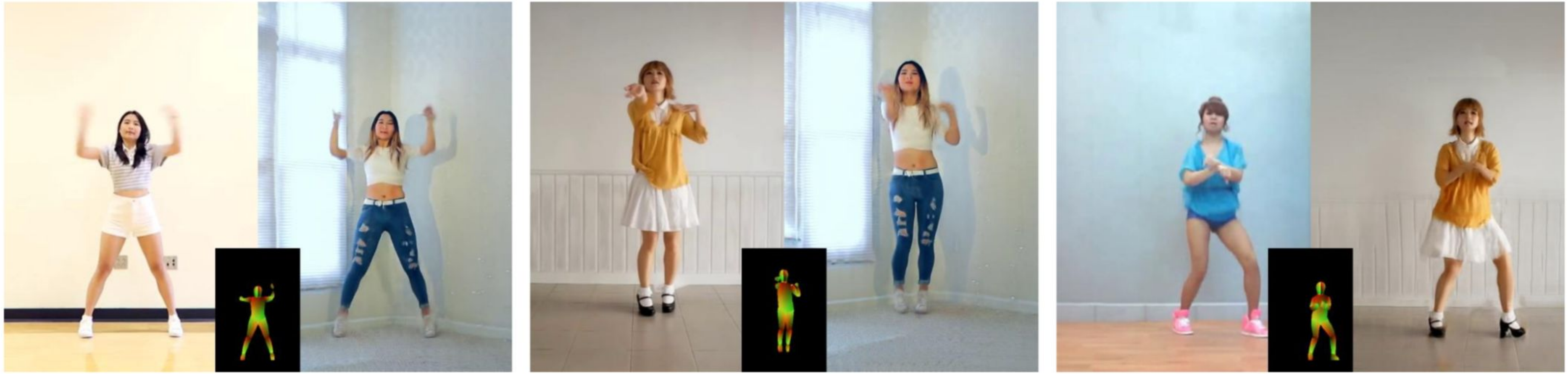
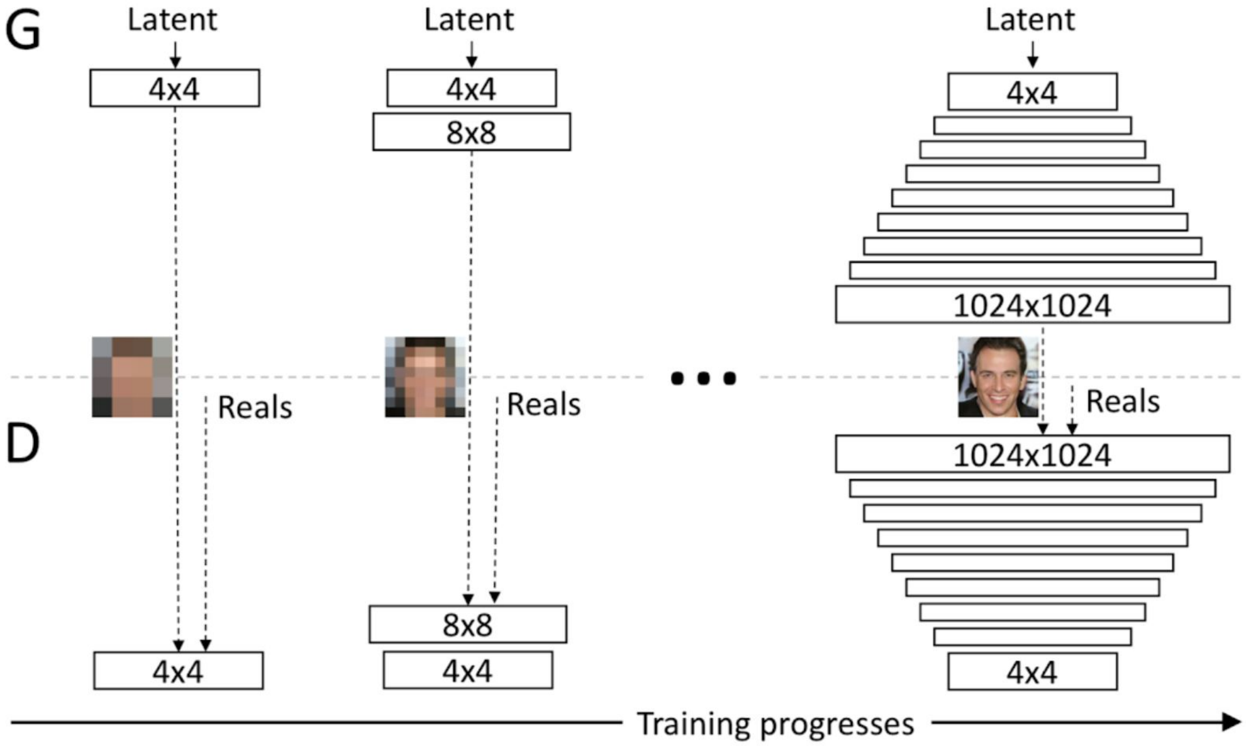


Figure 6: Example dance→pose→dance results. Each set shows the original dancer, the extracted poses, and the synthesized video. *Click the image to play the video clip in a browser.*

https://tcwang0509.github.io/vid2vid/paper_gifs/pose.mp4

Figure 7: Future video prediction results. Top left: ground truth. Top right: PredNet [45]. Bottom left: MCNet [68]. Bottom right: ours. *Click the image to play the video clip in a browser.* https://tcwang0509.github.i

Progressive Growing of GANs



https://openreview.net/forum?id=Hk99zCeAb

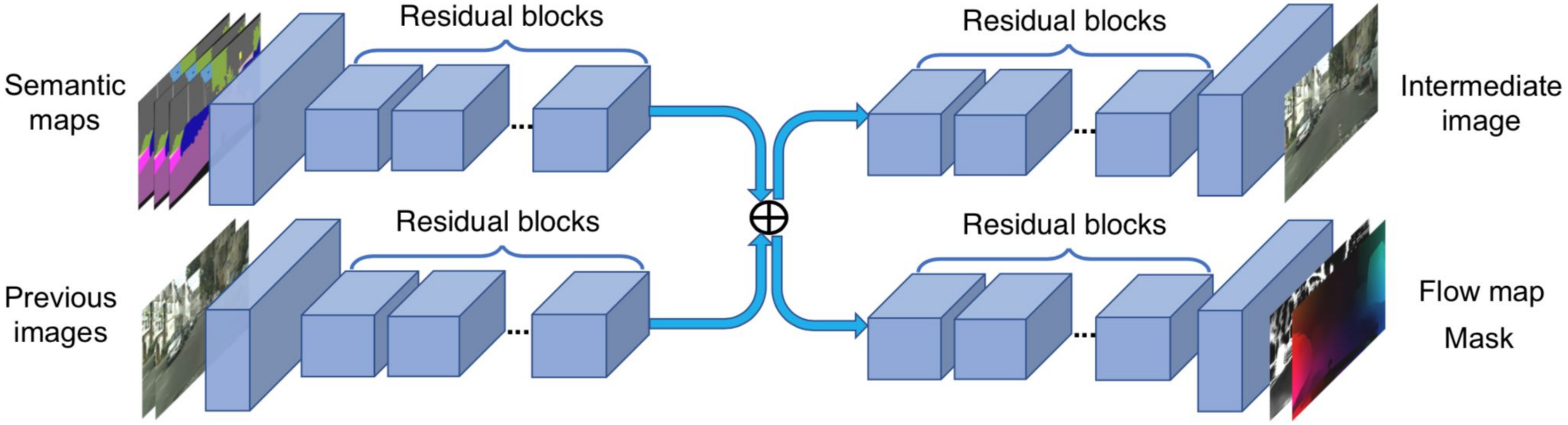


Figure 8: The network architecture (G_1) for low-res videos. Our network takes in a number of semantic label maps and previously generated images, and outputs the intermediate frame as well as the flow map and the mask.

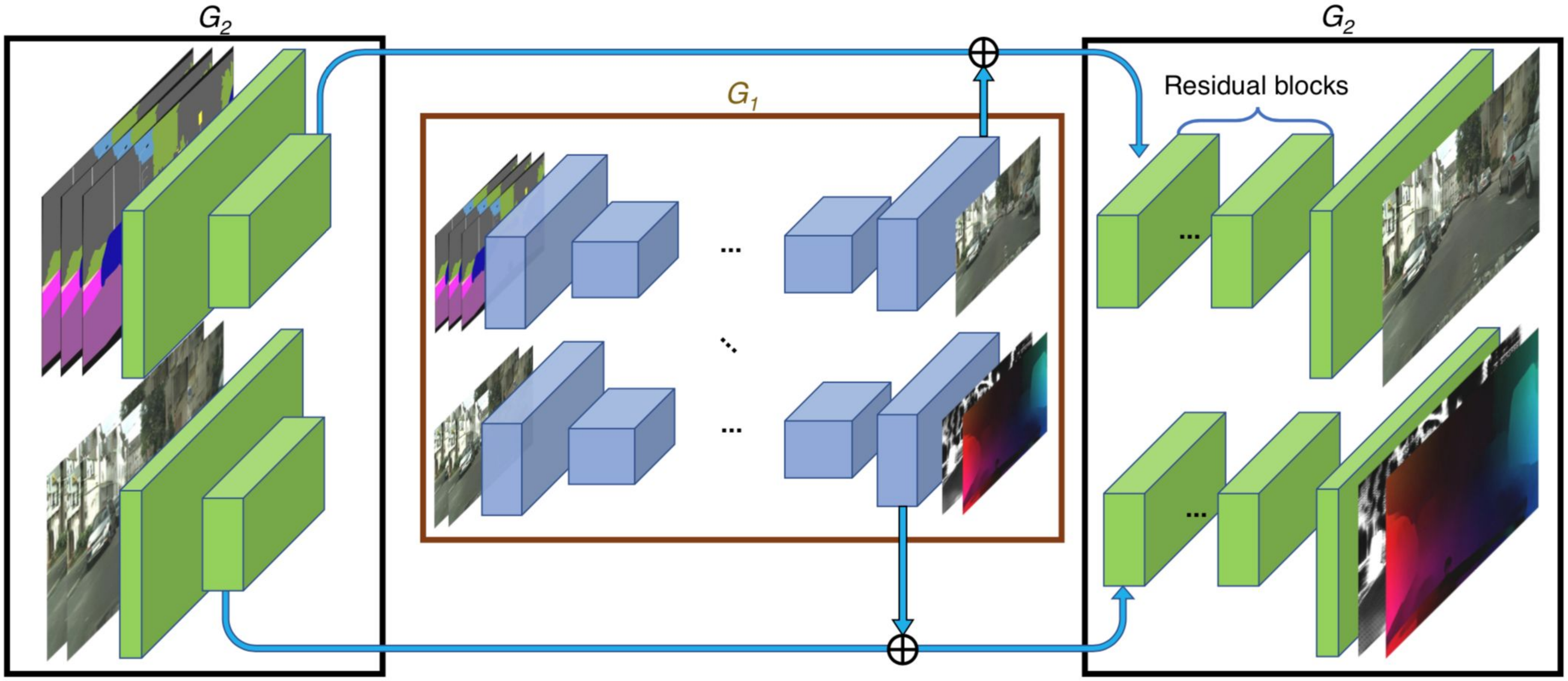


Figure 9: The network architecture (G_2) for higher resolution videos. The label maps and previous frames are downsampled and fed into the low-res network G_1. Then, the features from the high-res network and the last layer of the low-res network are summed and fed into another series of residual blocks to output the final images.

Interesting Papers & Trends NeurIPS 2018

- **Network architectures**
  - Perturbative NNs (CVPR 18 ;-)
  - Loss Function Visualization
  - Neural Differential Equations
- **GANs**
  - Vid2vid Video Generation
  - Text influenced Image Generation
- **Assorted Papers**
  - Dropblock Regularization
  - Features-Replay Parallel Training
  - DIfNet Semantic segmentation
- **Industry News**
  - Graphotate
  - Graphcore
  - News from NVidia
  - AI Report 2018

DropBlock: A regularization method for convolutional networks

Golnaz Ghiasi
Google Brain

Tsung-Yi Lin
Google Brain

Quoc V. Le
Google Brain

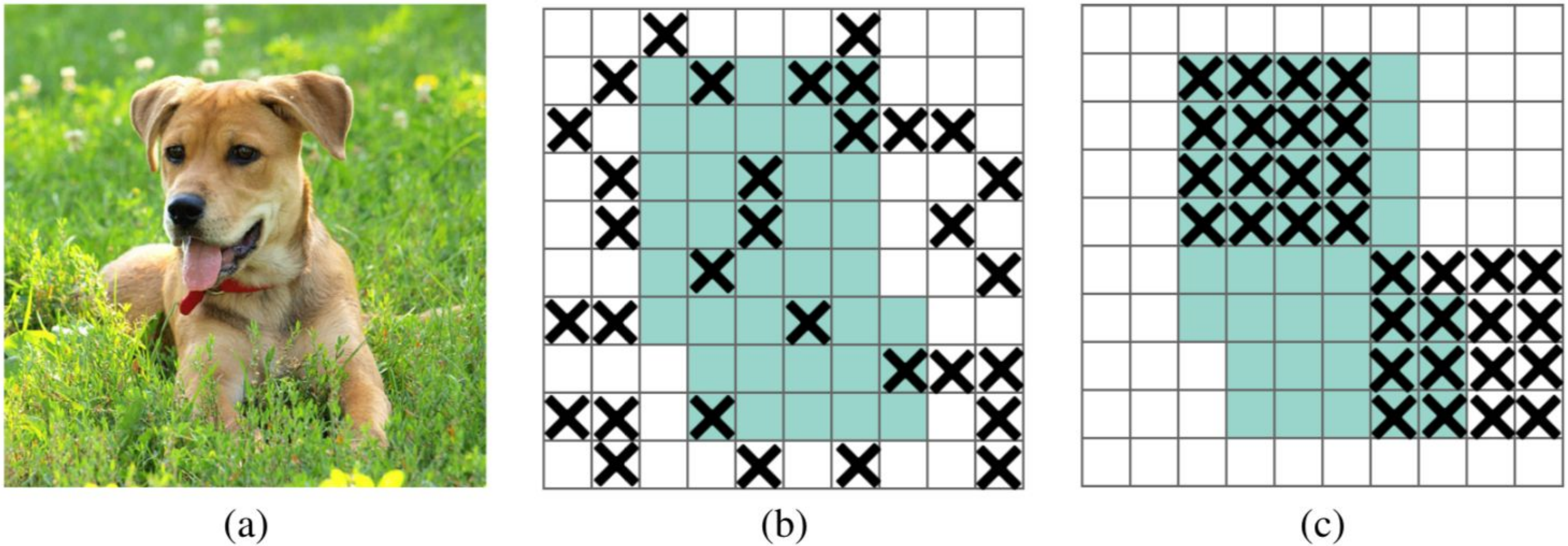
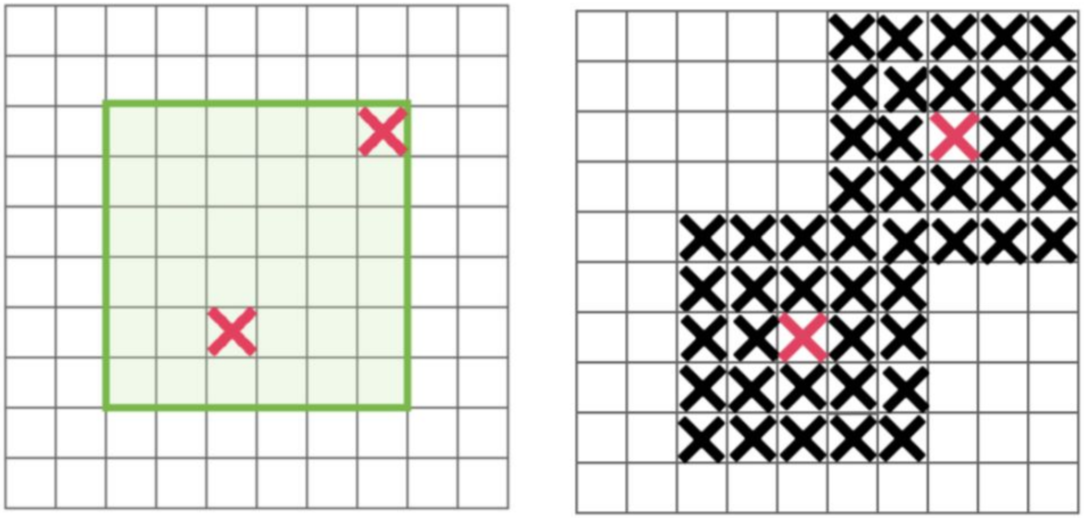


Figure 1: (a) input image to a convolutional neural network. The green regions in (b) and (c) include the activation units which contain semantic information in the input image. Dropping out activations at random is not effective in removing semantic information because nearby activations contain closely related information. Instead, dropping continuous regions can remove certain semantic information (e.g., head or feet) and consequently enforcing remaining units to learn features for classifying input image.



(a) (b)

Figure 2: Mask sampling in DropBlock. (a) On every feature map, similar to dropout, we first sample a mask M. We only sample mask from shaded green region in which each sampled entry can expanded to a mask fully contained inside the feature map. (b) Every zero entry on M is expanded to $block\_size \times block\_size$ zero block.

Model	top-1(%)	top-5(%)
ResNet-50	76.51 ± 0.07	93.20 ± 0.05
ResNet-50 + dropout (kp=0.7) [1]	76.80 ± 0.04	93.41 ± 0.04
ResNet-50 + DropPath (kp=0.9) [17]	77.10 ± 0.08	93.50 ± 0.05
ResNet-50 + SpatialDropout (kp=0.9) [20]	77.41 ± 0.04	93.74 ± 0.02
ResNet-50 + Cutout [23]	76.52 ± 0.07	93.21 ± 0.04
ResNet-50 + AutoAugment [27]	77.63	93.82
ResNet-50 + label smoothing (0.1) [28]	77.17 ± 0.05	93.45 ± 0.03
ResNet-50 + DropBlock, (kp=0.9)	78.13 ± 0.05	94.02 ± 0.02
ResNet-50 + DropBlock (kp=0.9) + label smoothing (0.1)	78.35 ± 0.05	94.15 ± 0.03

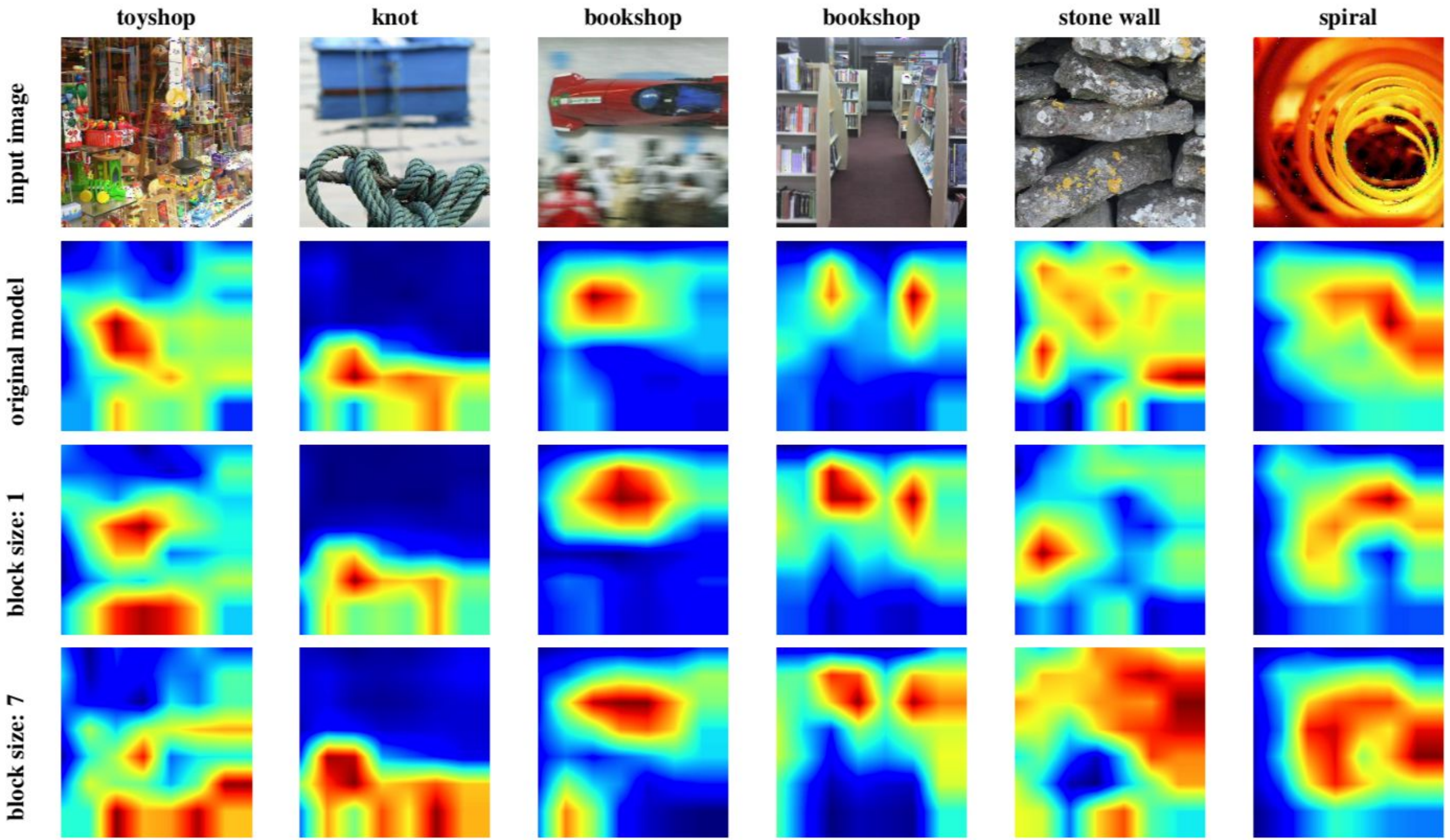
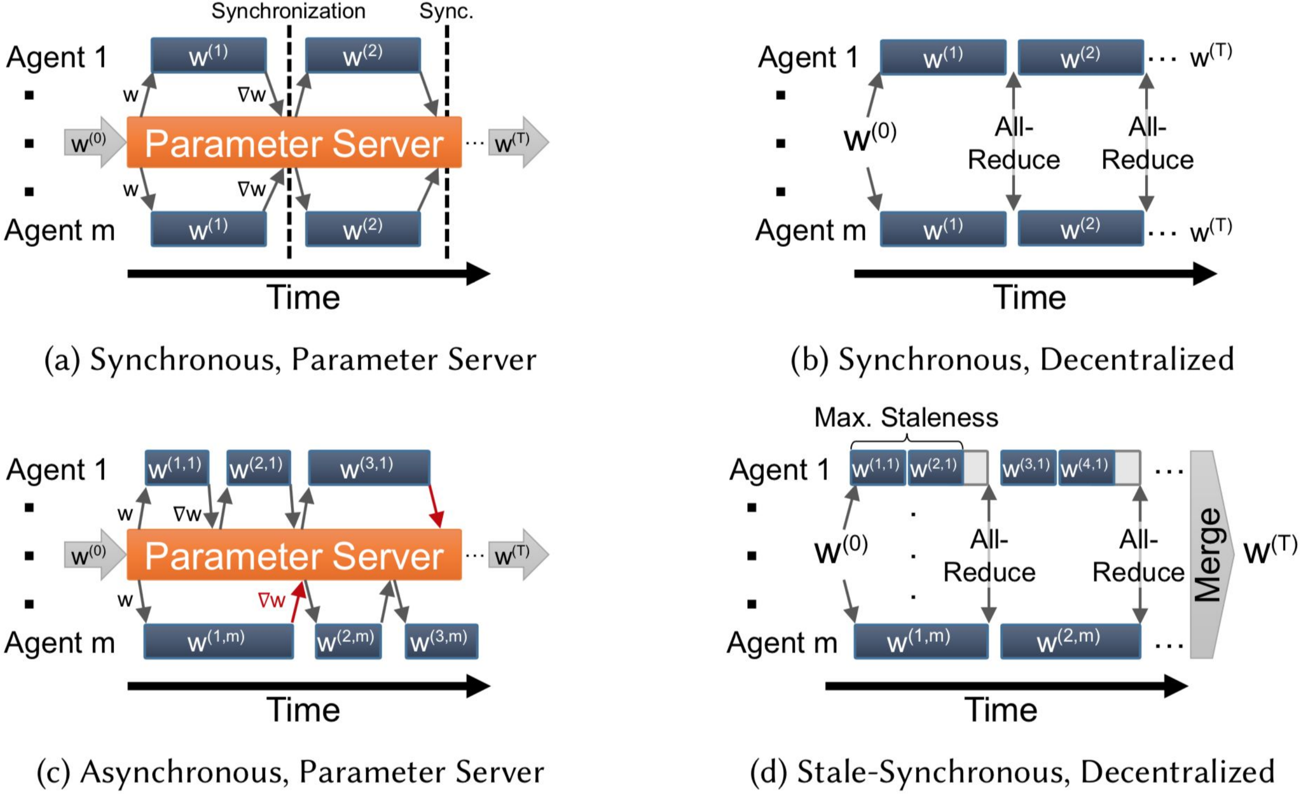


Figure 6: Class activation mapping (CAM) [29] for ResNet-50 model trained without DropBlock and trained with DropBlock with the $block\_size$ of 1 or 7. The model trained with DropBlock tends to focus on several spatially distributed regions.

Training Neural Networks Using Features Replay

Zhouyuan Huo [1], Bin Gu[2], Heng Huang[1,2*]
[1] Department of Electrical and Computer Engineering, University of Pittsburgh, USA, [2] JD.com
zhouyuan.huo@pitt.edu, jsgubin@gmail.com
heng.huang@pitt.edu



(a) Synchronous, Parameter Server

(b) Synchronous, Decentralized

(c) Asynchronous, Parameter Server

(d) Stale-Synchronous, Decentralized

Fig. 20. Training Distribution in Deep Learning (Model Consistency, Centralization)

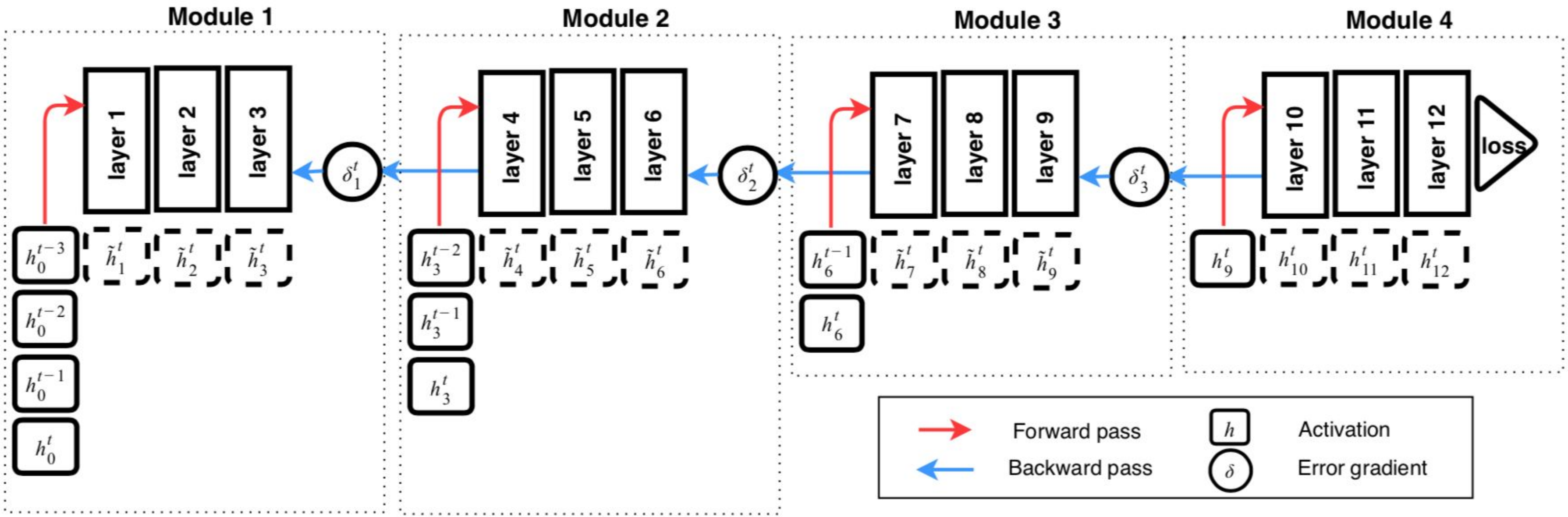
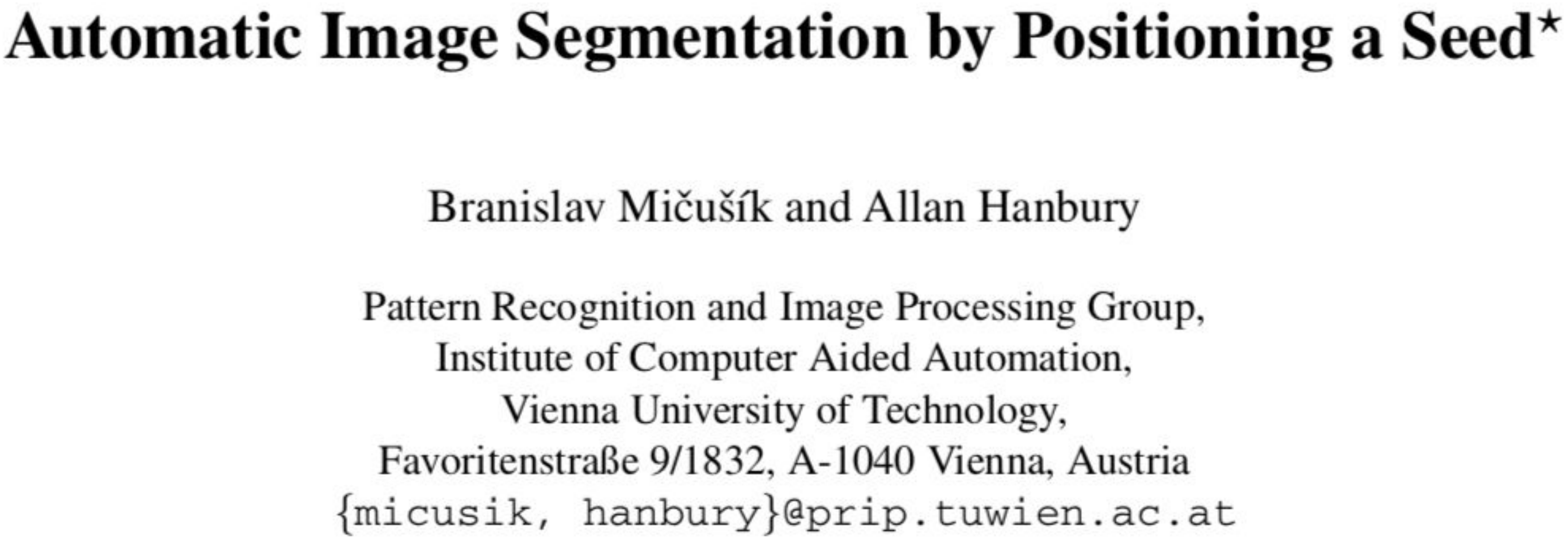


Figure 2: Backward pass of Features Replay Algorithm. We divide a 12-layer neural network into four modules, where each module stores its input history and a stale error gradient from the upper module. At each iteration, all modules compute the activations by inputting features from the history and compute the gradients by applying the chain rule. After that, they receive the error gradients from the upper modules for the next iteration.

Object / Semantic Segmentation

Object Segmentation – Graph Cuts (2006)


Automatic Image Segmentation by Positioning a Seed*

Branislav Mičušík and Allan Hanbury

Pattern Recognition and Image Processing Group,
Institute of Computer Aided Automation,
Vienna University of Technology,
Favoritenstraße 9/1832, A-1040 Vienna, Austria
{micusik, hanbury}@prip.tuwien.ac.at

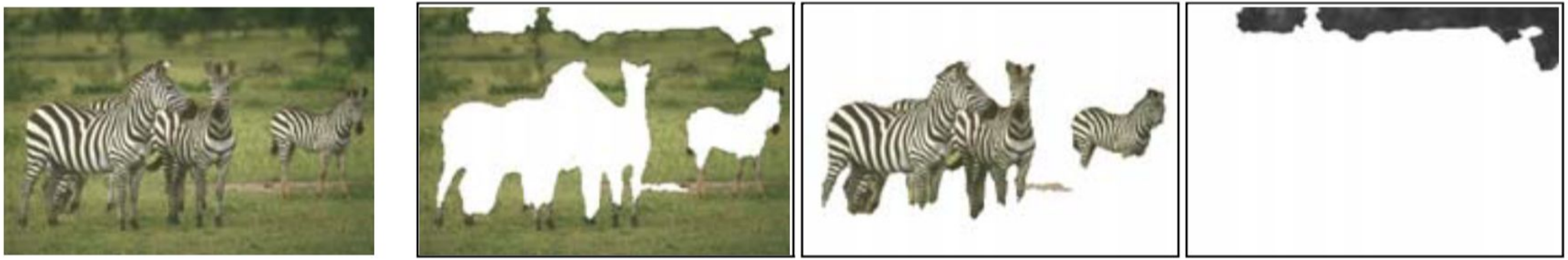
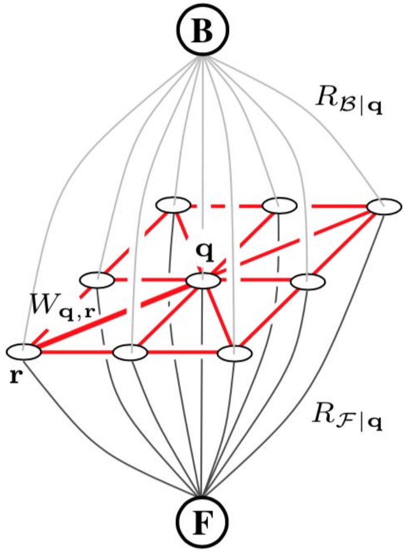


Fig. 1. Automatic segmentation of the zebra image shown at the left. The three images on the right show three dominant textures as three different regions produced by the proposed method.



edge	cost	region
$\{\mathbf{q},\mathbf{r}\}$	$W_{\mathbf{q},\mathbf{r}}$	$\{\mathbf{q},\mathbf{r}\} \in \mathcal{N}$
$\{\mathbf{q}, F\}$	$\lambda\, R_{\mathcal{F}\mid\mathbf{q}}$	$\forall \mathbf{q}$
$\{\mathbf{q}, B\}$	$\lambda\, R_{\mathcal{B}\mid\mathbf{q}}$	$\forall \mathbf{q}$

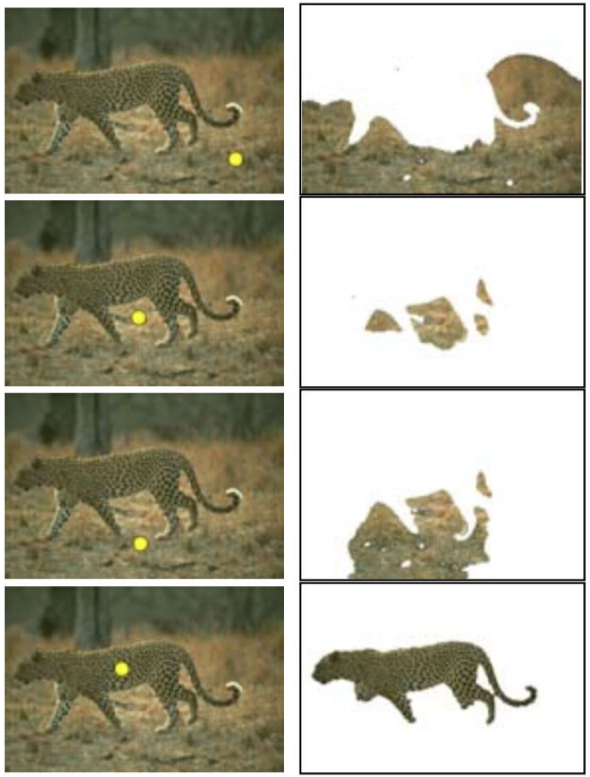
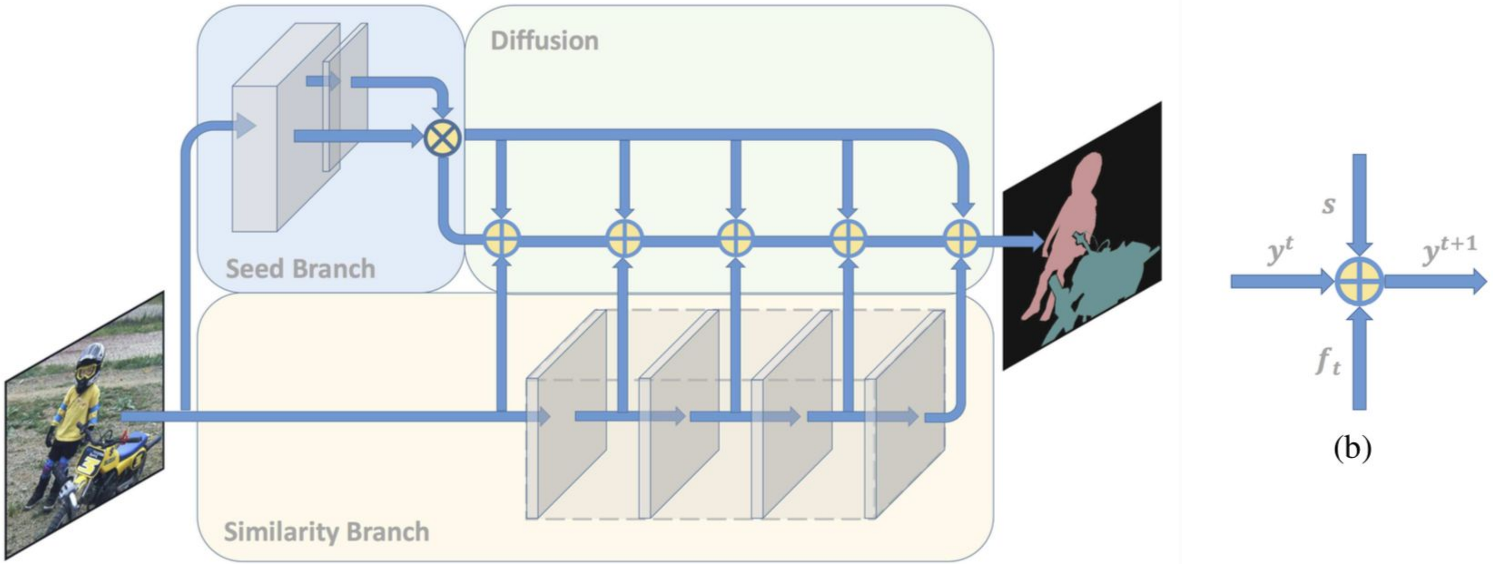


Fig. 2. Left: Graph representation for a 9 pixel image and a table defining the costs of graph edges. Symbols are explained in the text. Right: Four binary image segmentations using various positions of the seed.

DifNet: Semantic Segmentation by Diffusion Networks

Peng Jiang [1] **Fanglin Gu** [1] **Yunhai Wang** [1]
Changhe Tu [1] **Baoquan Chen** [2,1]
[1]Shandong University, China [2]Peking University, China
sdujump@gmail.com, fanglin.gu@gmail.com, cloudseawang@gmail.com
chtu@sdu.edu.cn, baoquan.chen@gmail.com



(a)

(b)

Figure 1: (a) Our DifNet contains two branches: 1. Seed Branch, which produces score map and importance map, from which seed are obtained by Hadamard product $\otimes$; 2. Similarity Branch, which extracts features from different layers to compute transition matrices and estimate pixel-wise similarities. Finally, the model approximates the diffusion process by a cascade of random walks $\oplus$ to propagate seed information to the whole image according to the estimated similarities. (b) Random Walk operation. For each random walk, the inputs are: 1. Output of last random walk operation; 2. Features from Similarity Branch; 3. Seed from Seed Branch. Given the inputs, output is calculated by $\oplus$. The specific computation procedure will be explained in Sec. 4.

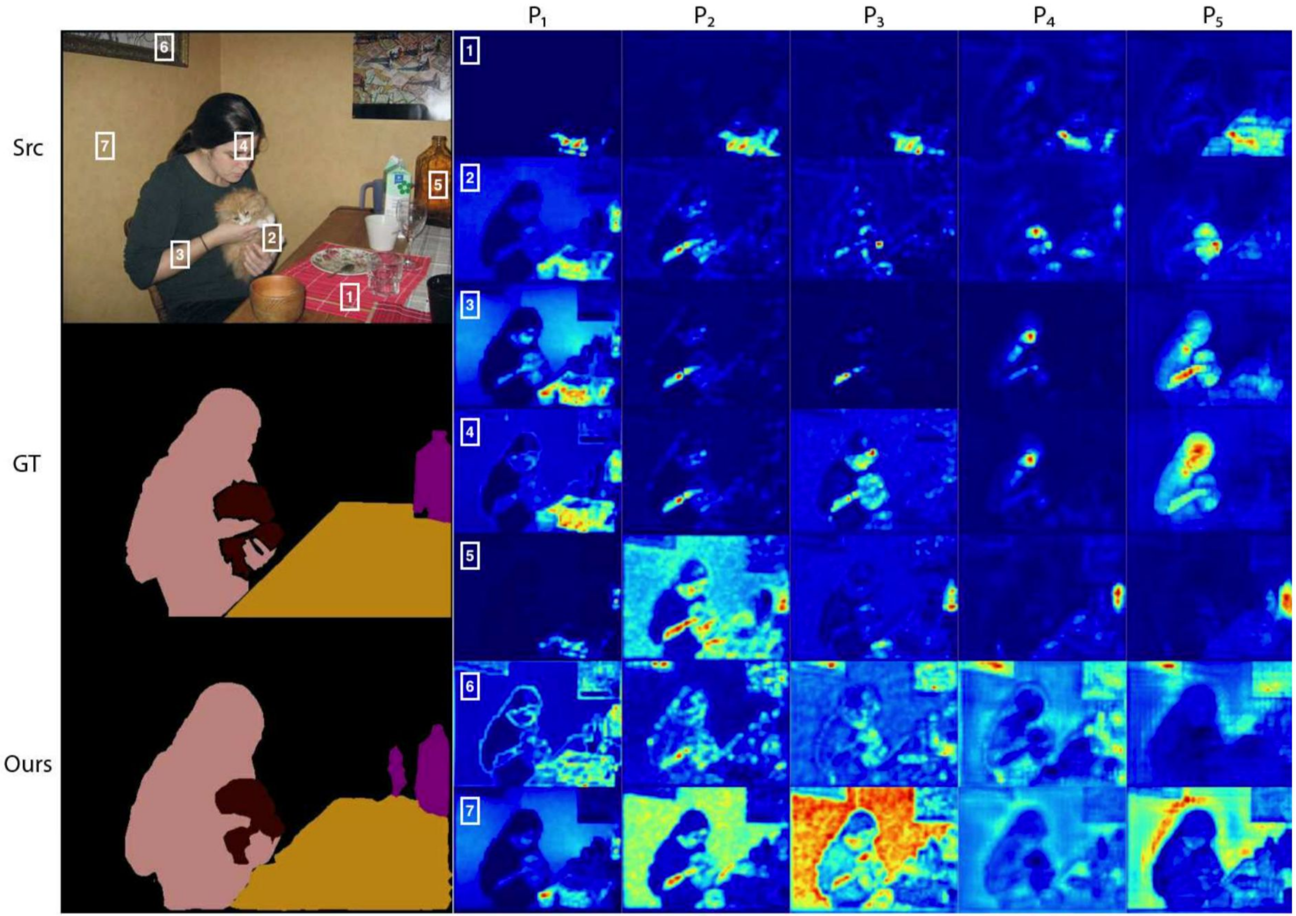


Figure 3: Visualization of corresponding rows in P_t of selected nodes in the image. The right five columns demonstrate similarities measured on different P_t, respectively. Nodes more highlighted are more similar to the selected node.

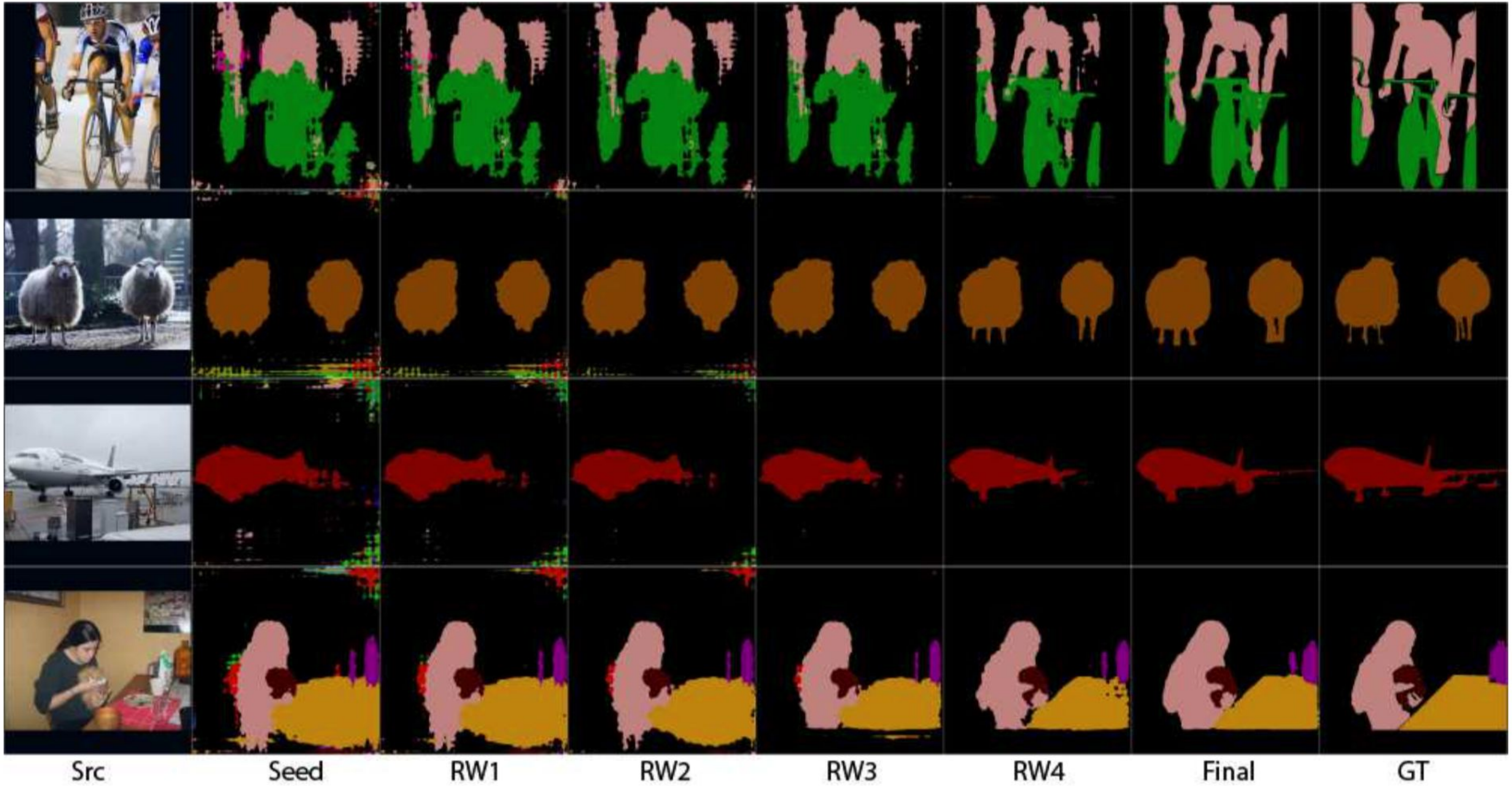
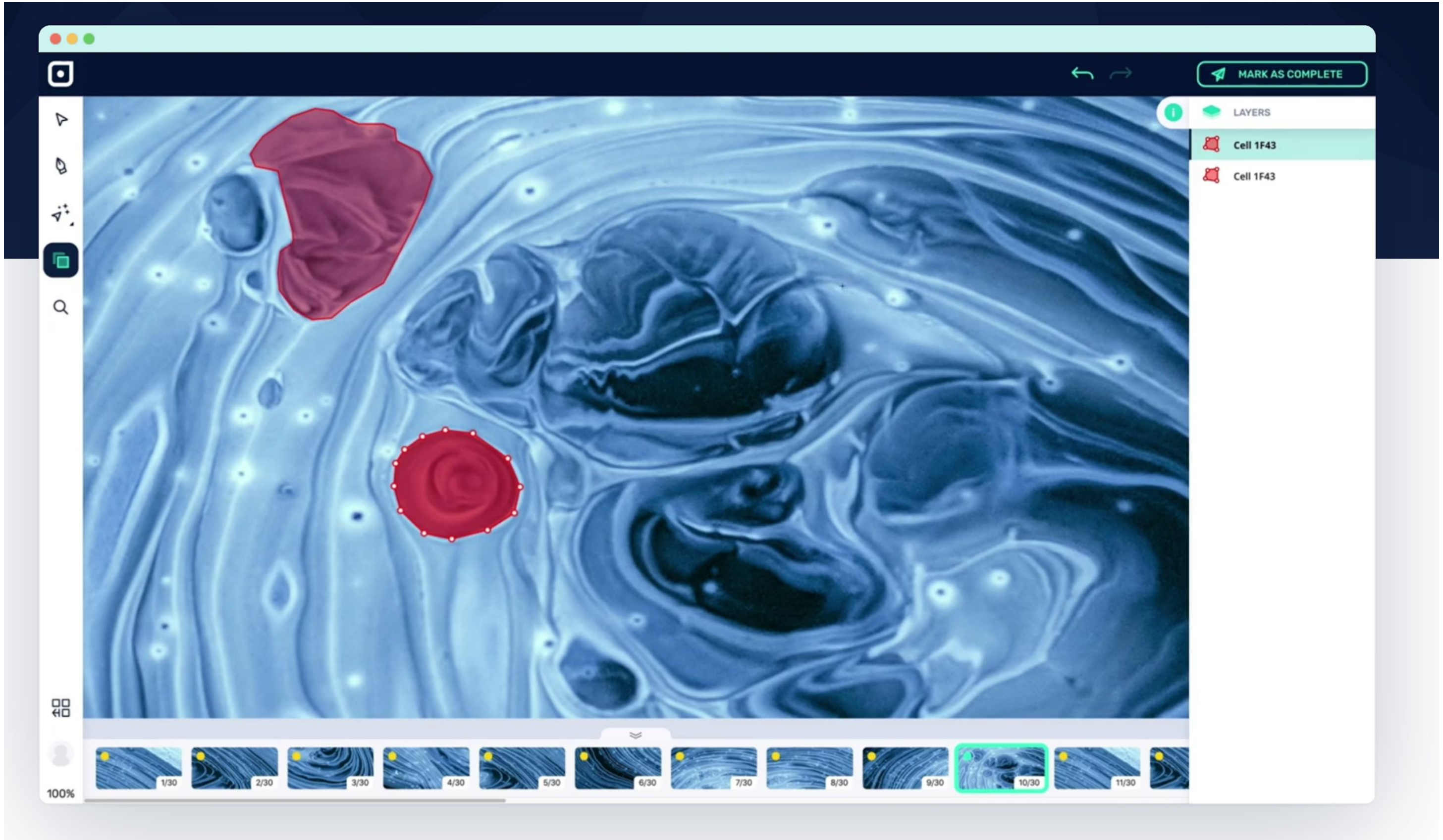


Figure 4: Visualization of our seed and outputs after each random walk $\oplus$.

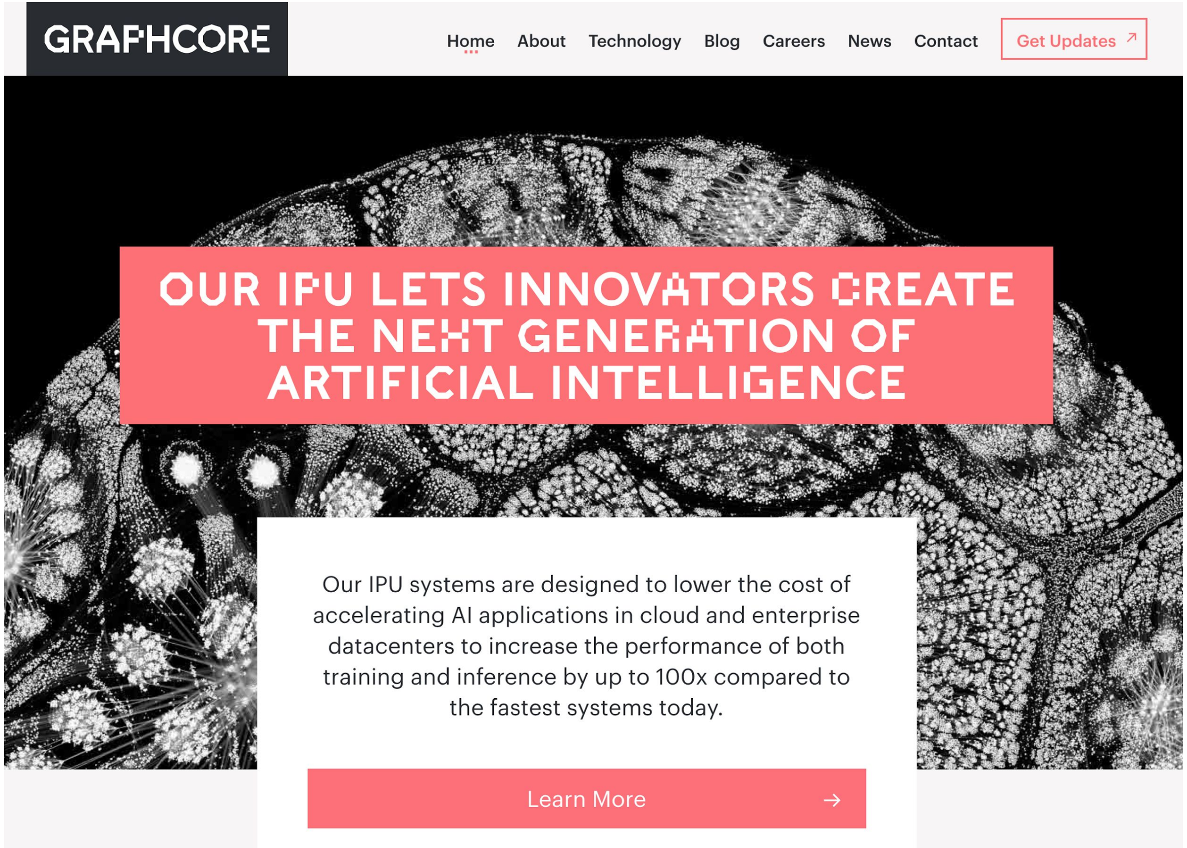
Interesting Papers & Trends NeurIPS 2018

- **Network architectures**
  - Perturbative NNs (CVPR 18 ;-)
  - Loss Function Visualization
  - Neural Differential Equations
- **GANs**
  - Vid2vid Video Generation
  - Text influenced Image Generation
- **Assorted Papers**
  - Dropblock Regularization
  - Features-Replay Parallel Training
  - DIfNet Semantic segmenation
- **Industry News**
  - Graphotate
  - Graphcore
  - News from NVidia
  - AI Report 2018

Graphotate


MARK AS COMPLETE
LAYERS
Cell 1F43
Cell 1F43
100%
1/30
2/30
3/30
4/30
5/30
6/30
7/30
8/30
9/30
10/30
11/30

Graphcore


GRAPHCORE
Home
About
Technology
Blog
Careers
News
Contact
Get Updates
OUR IPU LETS INNOVATORS CREATE THE NEXT GENERATION OF ARTIFICIAL INTELLIGENCE
Our IPU systems are designed to lower the cost of accelerating AI applications in cloud and enterprise datacenters to increase the performance of both training and inference by up to 100x compared to the fastest systems today.
Learn More

Graphcore

Graphcore's "Colossus"

(in honour of Tommy Flowers)

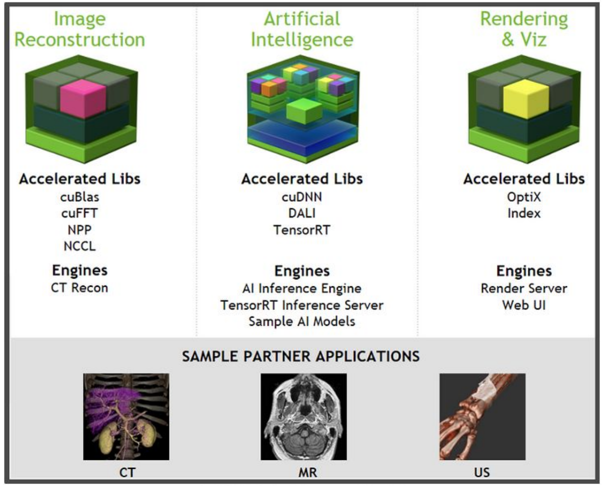
- Designed ground-up for MI, both training and deployment.
- Large 16nm custom chip, cluster-able, 2 per PCIe card.
- >1000 truly independent processors per chip; all-to-all non-blocking exchange.
- All model state remains on chip; no directly-attached DRAM.
- Mixed-precision floating-point stochastic arithmetic.
- DNN performance well beyond Volta and TPU2; efficient without large batches.
- Unprecedented flexibility for non-DNN models; thrives on sparsity.
- Program in TensorFlow initially, other o/s frameworks follow; or Poplar™ for close-to-metal.
- Early access cards and appliances end-2017.

News from NVidia

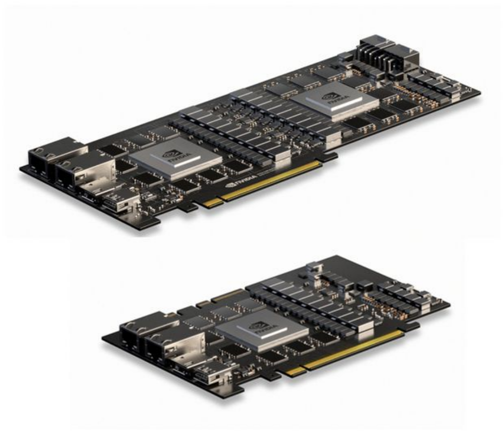
- A lot of research in the GAN / Automotive / Medical field
- TitanV with 32GB
- New graphics cards support 4bit and 1bit computations

- Clara / Rapids / DeepStream

The NVIDIA CLARA Platform



NVIDIA Clara SDK



NVIDIA Clara AGX

GE Healthcare
Cephasonics Ultrasound Solutions
MAYO CLINIC
ImFusion
NUANCE
KHEIRON MEDICAL TECHNOLOGIES
SUBTLE MEDICAL
VISAGE IMAGING®
aidence
THE OHIO STATE UNIVERSITY
ARTERYS
inferVISION 推想科技
imagia

NVIDIA Clara Developers

The RAPIDS suite of software libraries gives you the freedom to execute end-to-end data science and analytics pipelines entirely on GPUs. It relies on NVIDIA® CUDA® primitives for low-level compute optimization, but exposes that GPU parallelism and high-bandwidth memory speed through user-friendly Python interfaces.

RAPIDS also focuses on common data preparation tasks for analytics and data science. This includes a familiar DataFrame API that integrates with a variety of machine learning algorithms for end-to-end pipeline accelerations without paying typical serialization costs. RAPIDS also includes support for multi-node, multi-GPU deployments, enabling vastly accelerated processing and training on much larger dataset sizes.

RAPIDS Webpage

Intro Blog

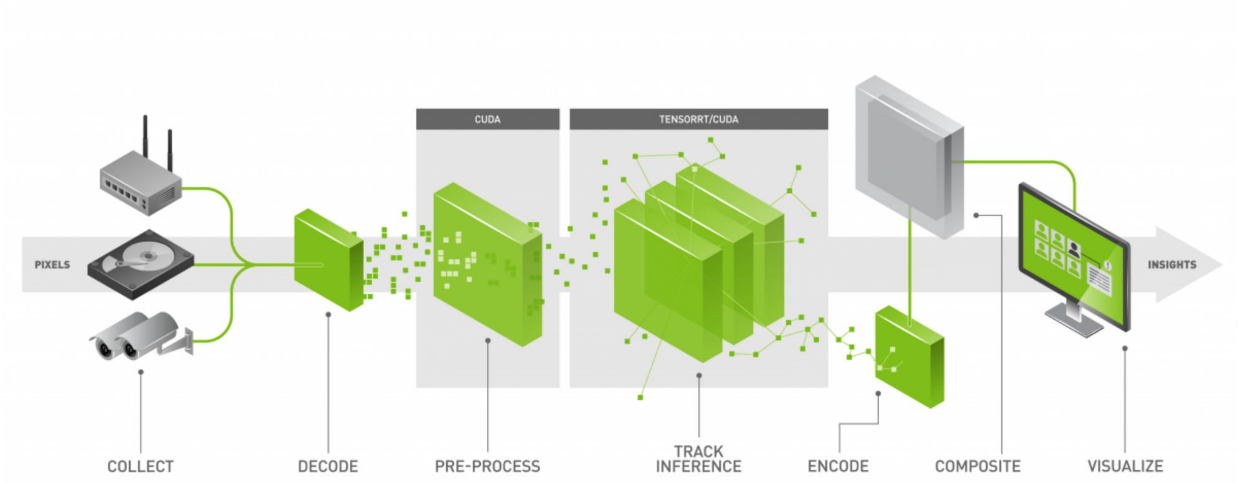
Features

Hassle-Free Integration Accelerate your Python data science toolchain with minimal code changes and no new tools to learn.	**Top Model Accuracy** Increase machine learning model accuracy by iterating on models faster and deploying them more frequently.
Reduced Training Time Drastically improve your productivity with near-interactive data science.	**Open Source** Customizable, extensible, interoperable - the open-source software is supported by NVIDIA and built on Apache Arrow.

Analyze Data From Cameras, Sensors and IoT Gateways in Real-Time

With more than a billion cameras and sensors continuously generating video streams and data, getting real-time actionable insights is more challenging than ever. NVIDIA's DeepStream SDK delivers a complete streaming analytics toolkit for situational awareness through intelligent video analytics (IVA) and multi-sensor processing.

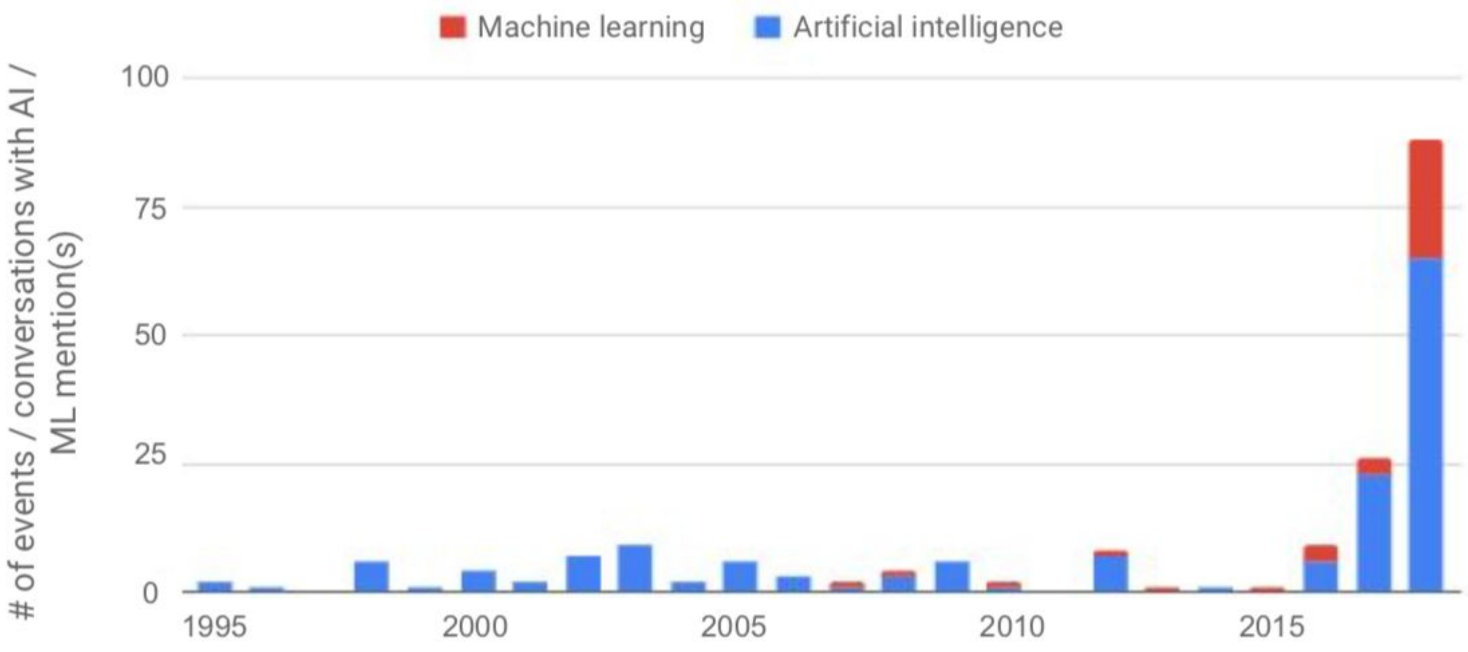
The DeepStream application framework features hardware-accelerated building blocks that bring deep neural networks and other complex processing tasks into a stream processing pipeline. Focus on building core deep learning networks and IP rather than designing end-to-end solutions from scratch.



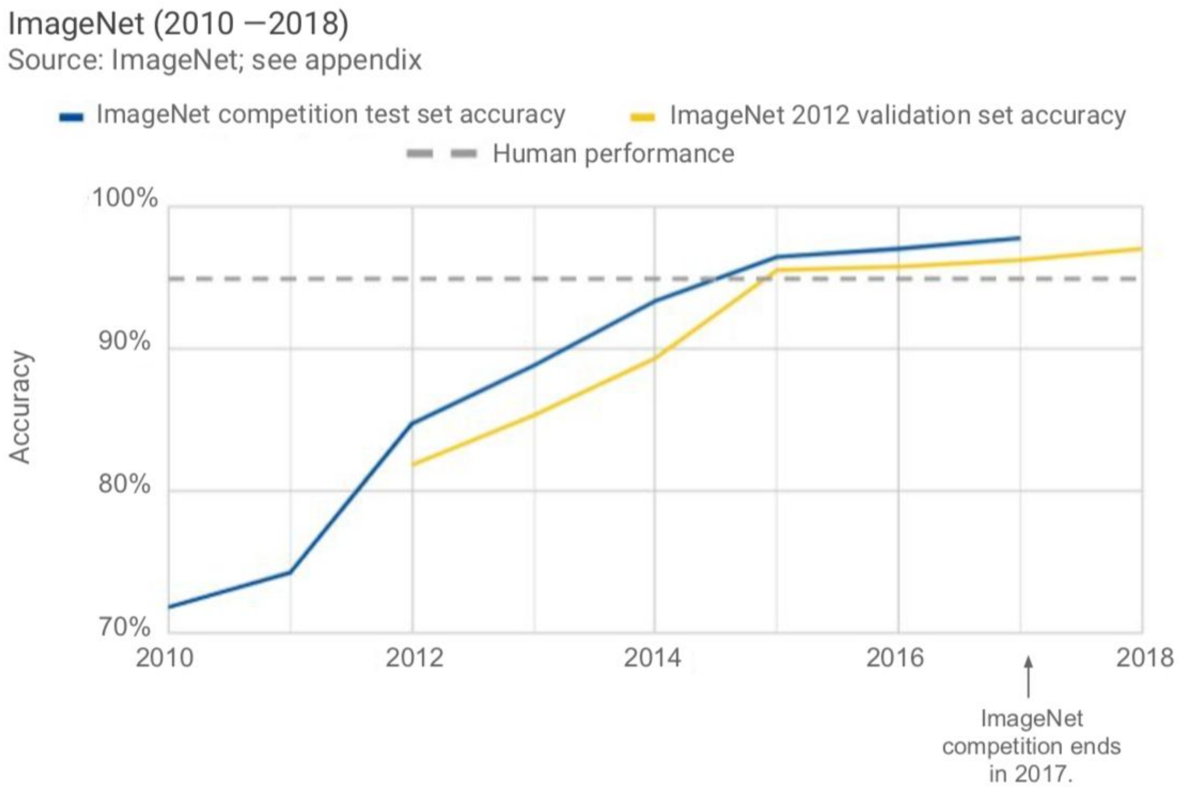
AI Index 2018 Report

artificial intelligence index
2018 annual report

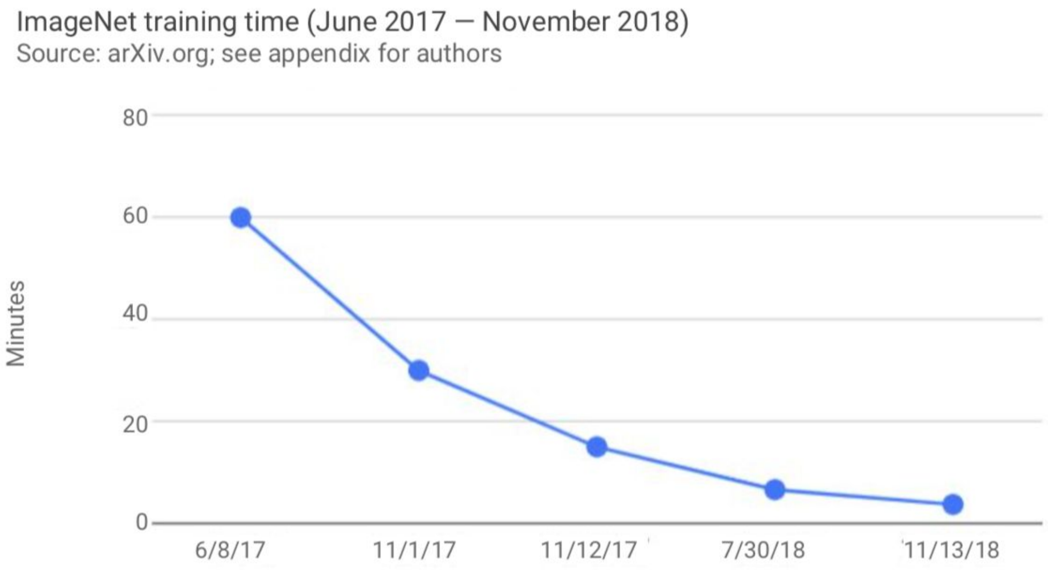
AI and ML mentions in U.S. Congress (1995—2018)
Source: U.S. Congressional Record website, McKinsey Global Institute analysis



AI Index 2018 Report

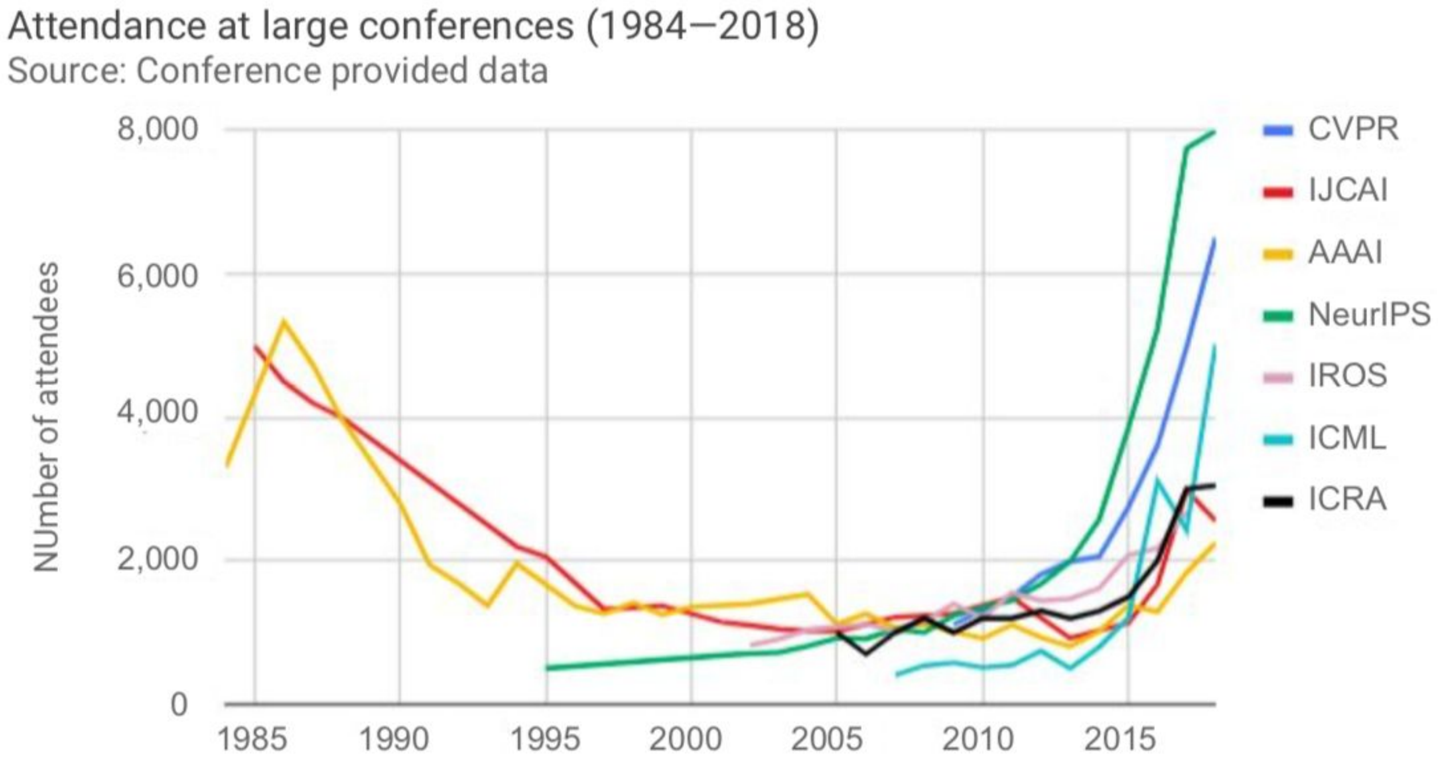


AI Index 2018 Report



ImageNet training time became 16x faster between June 2017 and November 2018

AI Index 2018 Report

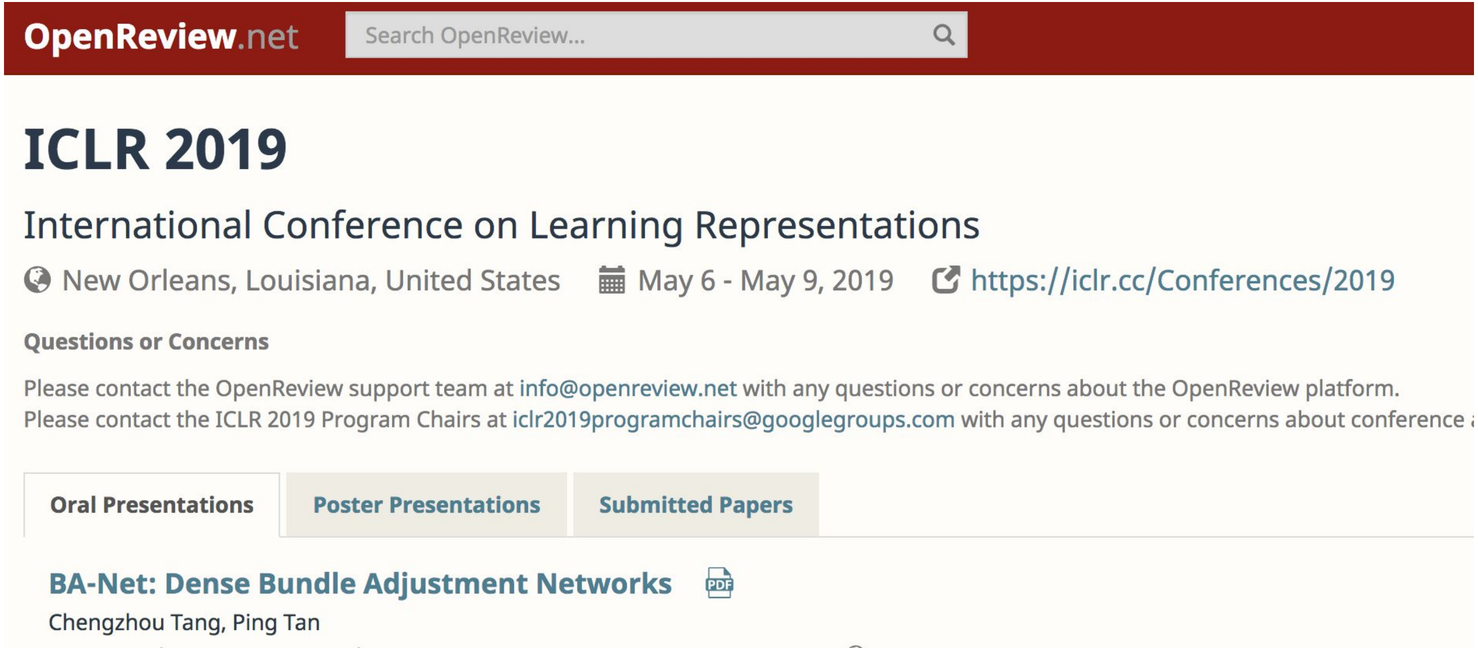


ICLR 2019 – New Orleans

https://iclr.cc/

Mon May 6th through Thu the 9th

Registration opens this Saturday, 1am (!) CET (4pm PST)



Future Topics Brainstorm

Future Topics Brainstorm

sli.do
#VDLM23

→ Ideas → Future Meetup Topics

Vienna
Deep Learning
Meetup

Vienna
Deep Learning
Meetup

Next Meetup:
28 February 2019
T-Mobile

Vienna
Deep Learning
Meetup