TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

AUSTRIAN INSTITUTE OF TECHNOLOGY

Deep Learning for Stereo Vision – Guest lecture

LVA Stereo Vision (188.513)
Summer Semester 2021
Andreas Kriegler

Modalities

- Contacts:
  - andreas.kriegler@tuwien.ac.at
  - margrit.gelautz@tuwien.ac.at

- You can raise your (virtual) hand or speak freely to ask a question
- Slides will be available in the TUWEL course

- Machine learning builds on decades of mathematical frameworks – here it is packed into ~30mins
- Some of you will not have acquired the prerequisite knowledge yet, so you may struggle to understand everything as we go – that‘s okay ☺

If you want to help in keeping the number of errors at a reasonable level send any that you find to andreas.kriegler@tuwien.ac.at

Literature

- ML & DL use applied statistics, linear algebra & calculus (books):
  - Mathematical foundations and many common algorithms of machine learning – the ML “bible”: [1]
  - The application of deep learning in neural networks: [2], [3]
  - Artificial intelligence in general and multi-agent theory: [4]
  - The necessity of statistics for robotics applications – probabilistic robotics: [5]
- Lectures:
  - TU Wien - 194.100 Theoretical Foundations and Research Topics in Machine Learning [6]
  - Stanford - CS229 Machine Learning [7]
  - Stanford - CS231n Convolutional Neural Networks for Visual Recognition [8]
  - MIT - Deep Learning and Artificial Intelligence Lectures [9]
- Videos:
  - Deep Learning Series, 3Blue1Brown [10]
  - Mathematics for Machine Learning, Ulrike von Luxburg [11]

Topics covered

I. Introduction to machine learning (ML) & deep learning (DL) ~15mins

II. Convolutional neural networks (CNN) ~10mins

III. Deep learning for stereo vision ~5mins

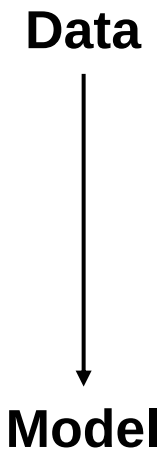
IV. Recent deep stereo vision methods ~55mins

V. If time allows it: detour to transformers ~5mins

Machine learning basics

- In classical programming we define rules for input → output relations

- In ML we use data to
  - 1. Generate our model (**learning**)
  - 2. Apply it to new observations (**inference**/prediction)



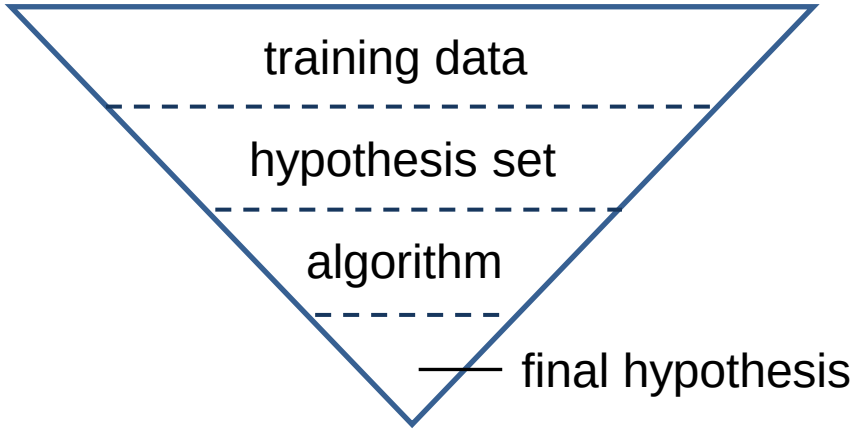


Figure recreated from [28]

Machine learning basics

- ML often based on calculating the **gradient** of a (convex) **loss-function**
- „No magic“ in ML/DL: find **minima** of objective/cost/error/target/loss function
- Try to find the **weights (parameters)** of the model that minimizes the cost

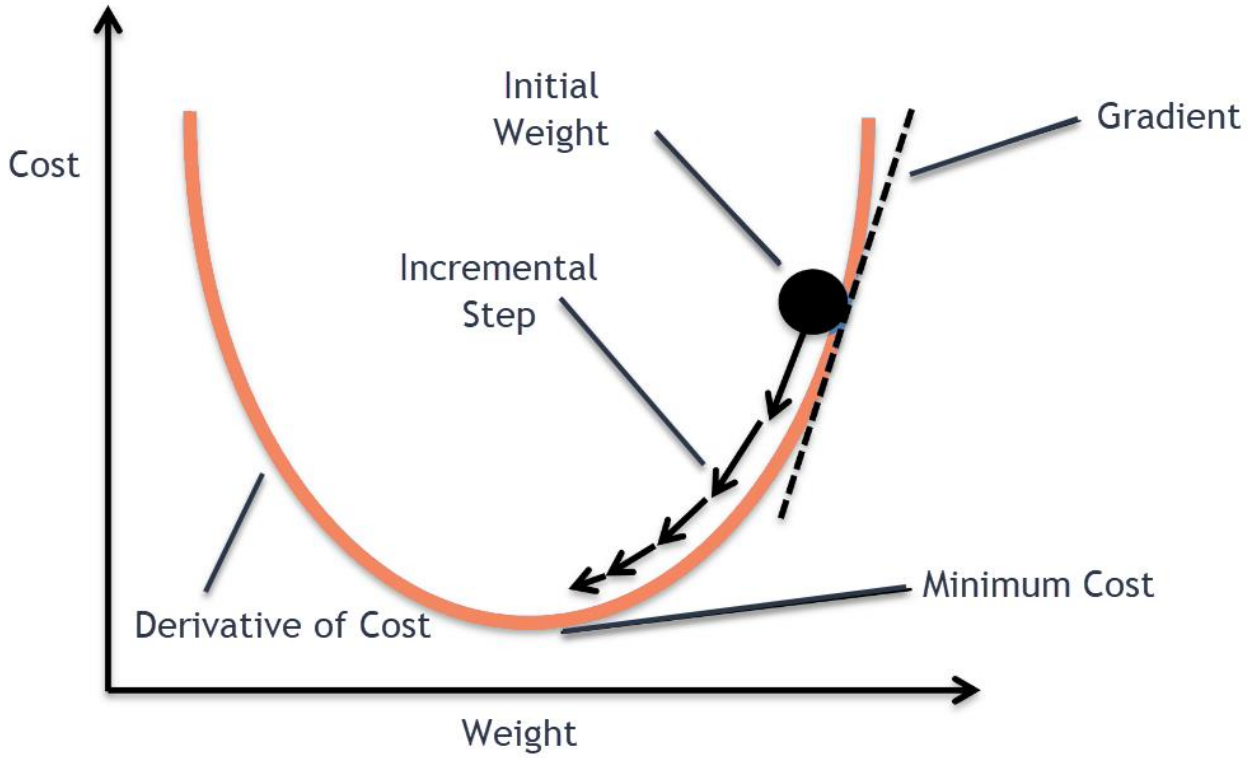


Figure taken from [27]

ML basics – supervised classification

ML example transcribed from [1]

- We have: a **training set** of N observations of x, $\boldsymbol{x} := (x_1, \dots, x_N)^\mathsf{T}$ with corresponding **target** observations t, $\boldsymbol{t} := (t_1, \dots, t_N)^\mathsf{T}$
- We want: predict $\hat{t}$ for new $\hat{x}$, using parameters $\boldsymbol{\theta}$
- **Regression** if $\boldsymbol{t} \subset \mathbb{R}$ or n-way **classification** if $\boldsymbol{t}$ is categorical $t \in \{0, \dots, n-1\}$
- We can, for example, try to fit a polynomial curve

$$f = y(x, \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_M x^M = \sum_{j=0}^{M} \theta_j x^j \qquad (1)$$

- We can calculate any loss/error function often L1 or L2

$$L1(\boldsymbol{\theta}) = \frac{1}{2} \sum_{n=1}^{N} \{|y(x_n, \boldsymbol{\theta}) - t_n|\} = ||\boldsymbol{\theta}||_1 \qquad (2)$$

$$L2(\boldsymbol{\theta}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \boldsymbol{\theta}) - t_n\}^2 = ||\boldsymbol{\theta}||_2 \qquad (3)$$

ML basics – supervised classification

ML example transcribed from [1]

$$L(\boldsymbol{\theta}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \boldsymbol{\theta}) - t_n\}^2 \quad (4)$$

- This will **overfit**, i.e. perform well on training but poorly on test samples
- To prevent this we add a term (λ) for **regularization**

$$\tilde{L}(\boldsymbol{\theta}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \boldsymbol{\theta}) - t_n\}^2 + \frac{\lambda}{2}||\boldsymbol{\theta}||^2 \quad (5)$$

- Now we can update – **optimize** – our parameters $\boldsymbol{\theta}$ with e.g. curve fitting methods

This was a simple example to introduce common terminology - in typical ML problems we want a convex loss function so we can use **gradient descent** techniques [26]

In ML the hat $\hat{x}$ is typically used to denote targets and the snake $\tilde{x}$ for estimates of some variable x

Deep learning – artificial neural networks

- Deep learning with multilayer perceptrons (MLP) since 1965 [12]
- Dealing with vanishing gradients since 1991 [13, 14], deep since 2012 [15]
- Now models with up to 10^{12} parameters trained on cloud-based tensor or graphical processing unit (TPU/GPU) clusters [16]

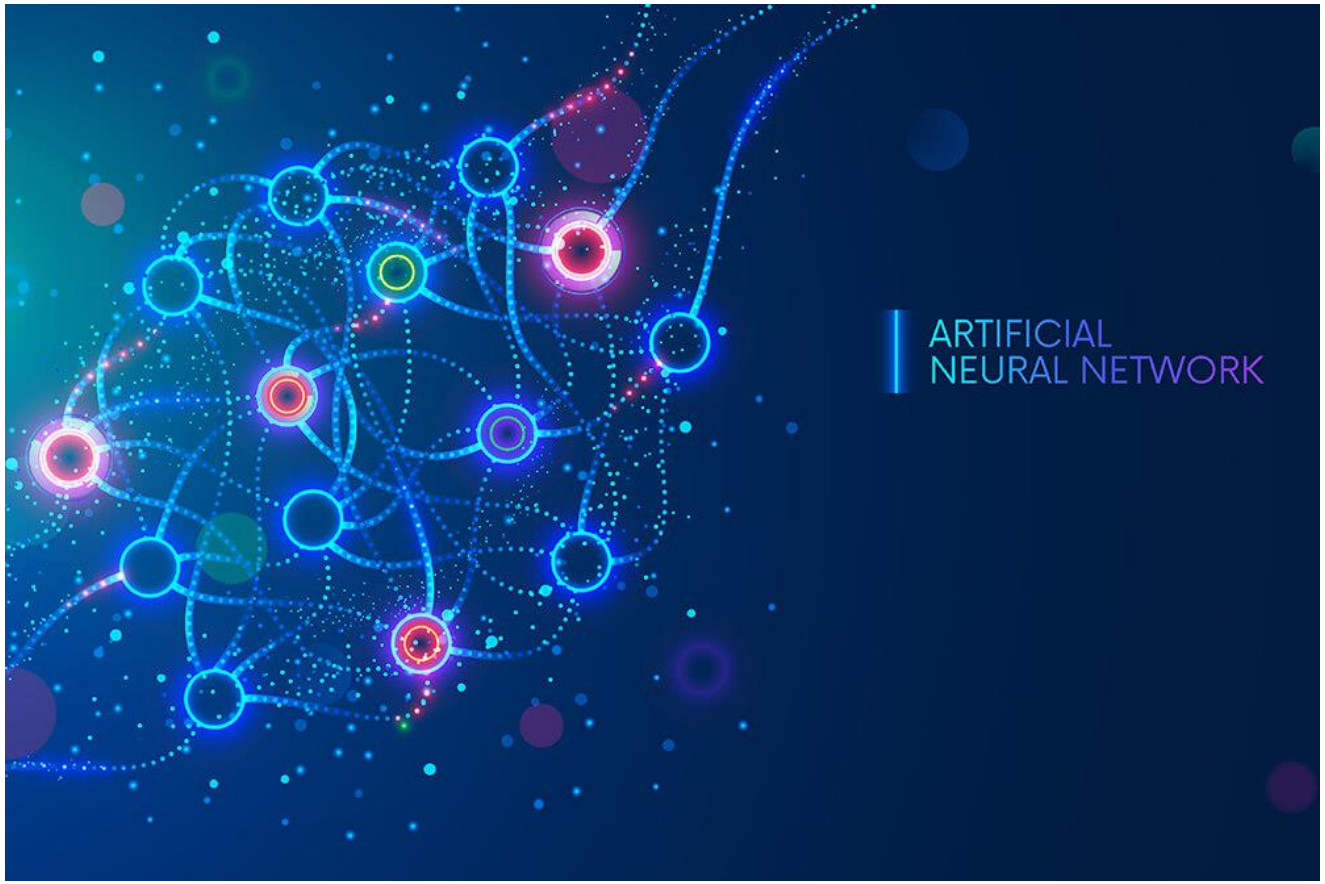
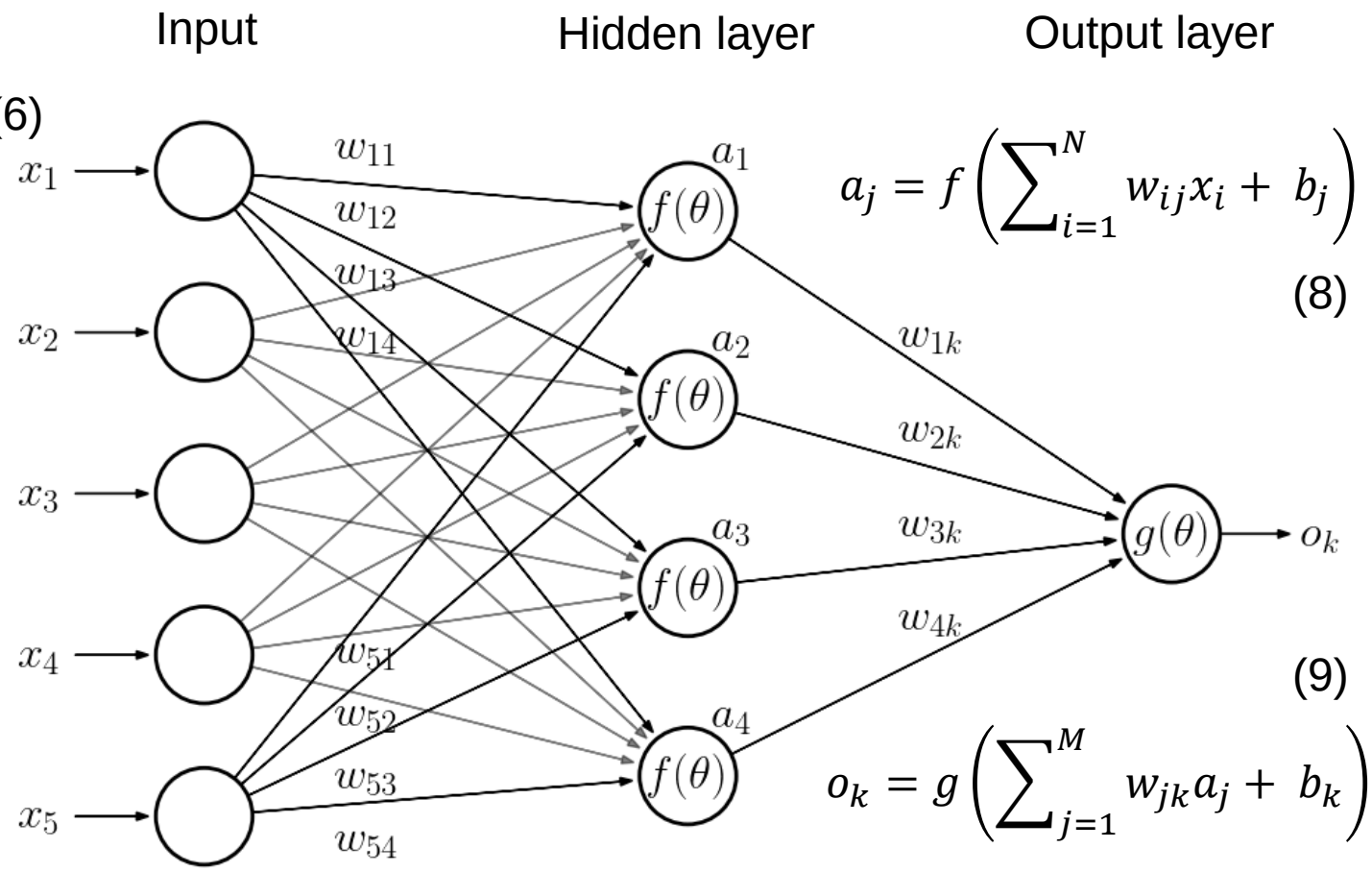


Figure taken from [29]

Deep learning – 1-hidden layer neural network

- **Activations** f in hidden neurons
  - Sigmoid (old): $\phi(z) = \frac{1}{1+e^{-z}}$ (6)
  - **ReLU** (rectified linear unit): $\mathrm{ReLU}(z) = \max(0, z)$ (7)
- **Weights** w scale the function and **bias** b shifts it
- N, M: number of input neurons – here 5 and 4
- Multiple output neurons form a output vector



$$a_j = f\left(\sum_{i=1}^{N} w_{ij}x_i + b_j\right) \quad (8)$$

$$o_k = g\left(\sum_{j=1}^{M} w_{jk}a_j + b_k\right) \quad (9)$$

Figure taken from [17]

Training NNs – obtaining the loss

- The output vector of the NN is squashed to predictions $\hat{y} \in (0, 1]$ using **softmax**

$$\hat{y} = \sigma(\boldsymbol{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{10}$$

- This „distribution over categorials“ allows probabilistic reasoning [74]
- We apply **cross-entropy loss** (simplified version) with $\hat{\boldsymbol{y}}$ and ground truth (gt) $\boldsymbol{y}$

$$L = -\boldsymbol{y} \cdot \log(\hat{\boldsymbol{y}}) \tag{11}$$

- NNs are typically updated using a variation of gradient descent: stochastic gradient descent (**SGD**)

Training NNs – gradient descent

- Stochastic gradient descent (**SGD**)

$$\boldsymbol{\theta} \coloneqq \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \tilde{L}(\boldsymbol{x}; \boldsymbol{t}) \quad \eta \ \ldots \text{ learning rate} \tag{12}$$

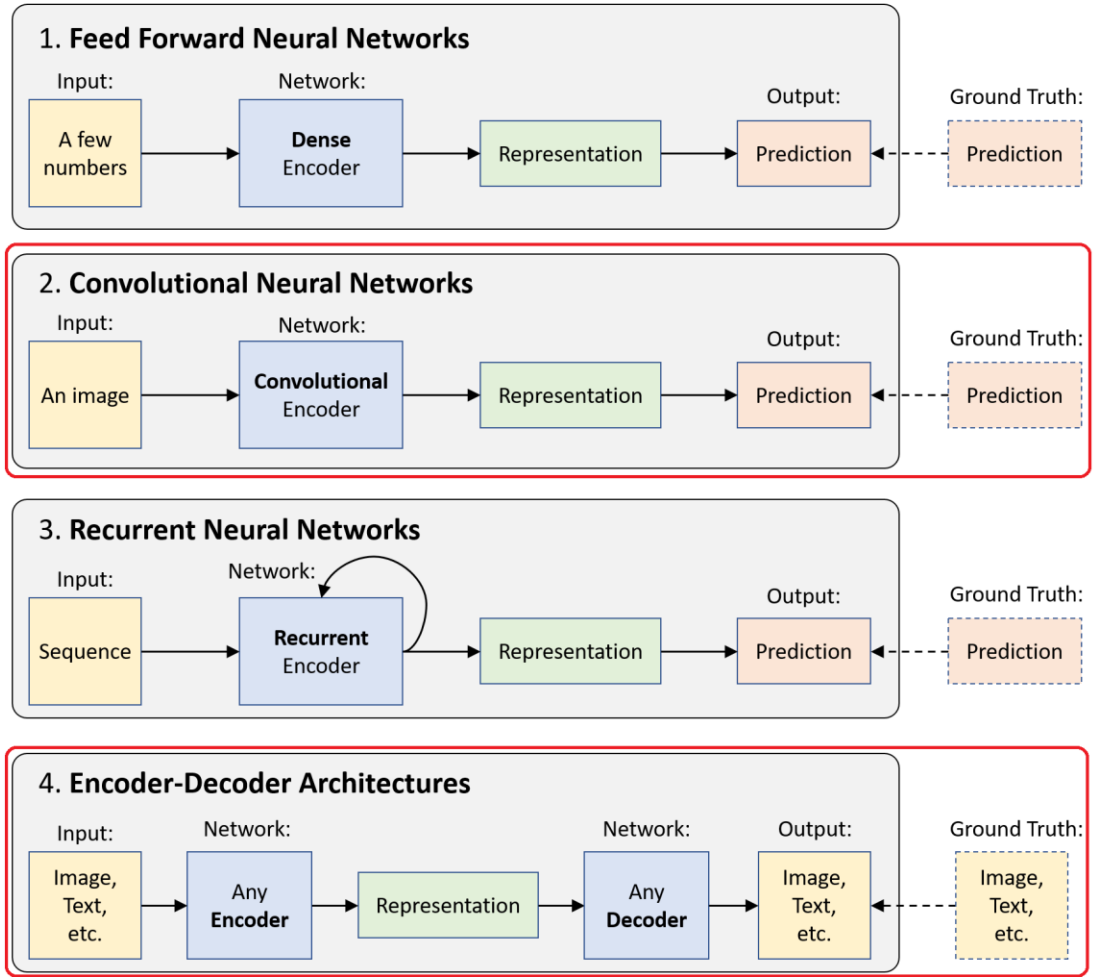
- Instead of the entire dataset, we only compute the gradient for small sub-sets
- Configuring η (as well as any other model hyperparameters) is its own science
- Backpropagation as a special case of (reverse) automatic differentiation [18] – based on Euler-LaGrange equations [19]

Check out the following 3Blue1Brown video (especially 10:30 – 11:26)

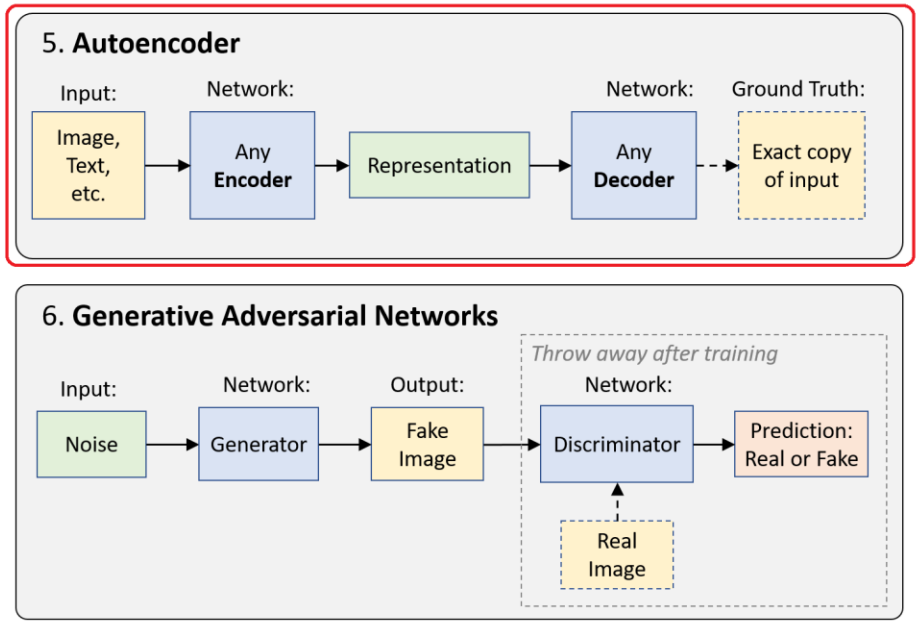
https://preview.tinyurl.com/m92b8r9u

Learning families and neural networks

Supervised Learning



Unsupervised Learning



Reinforcement Learning

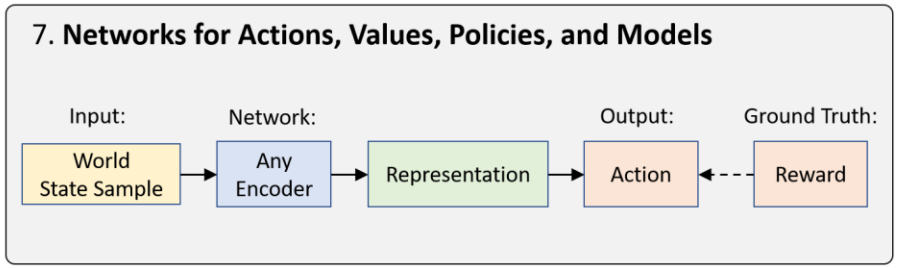


Figure slightly modified and taken from [9]

Supervised CNNs

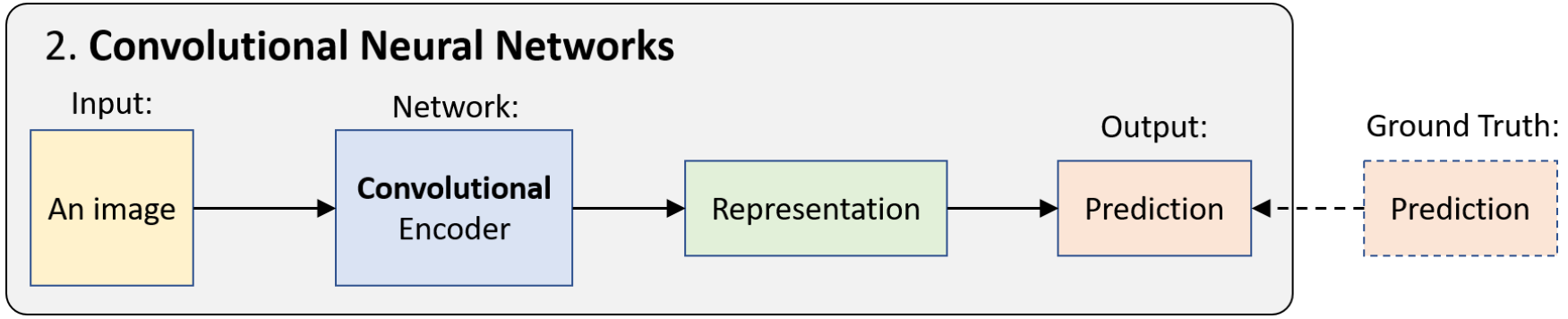
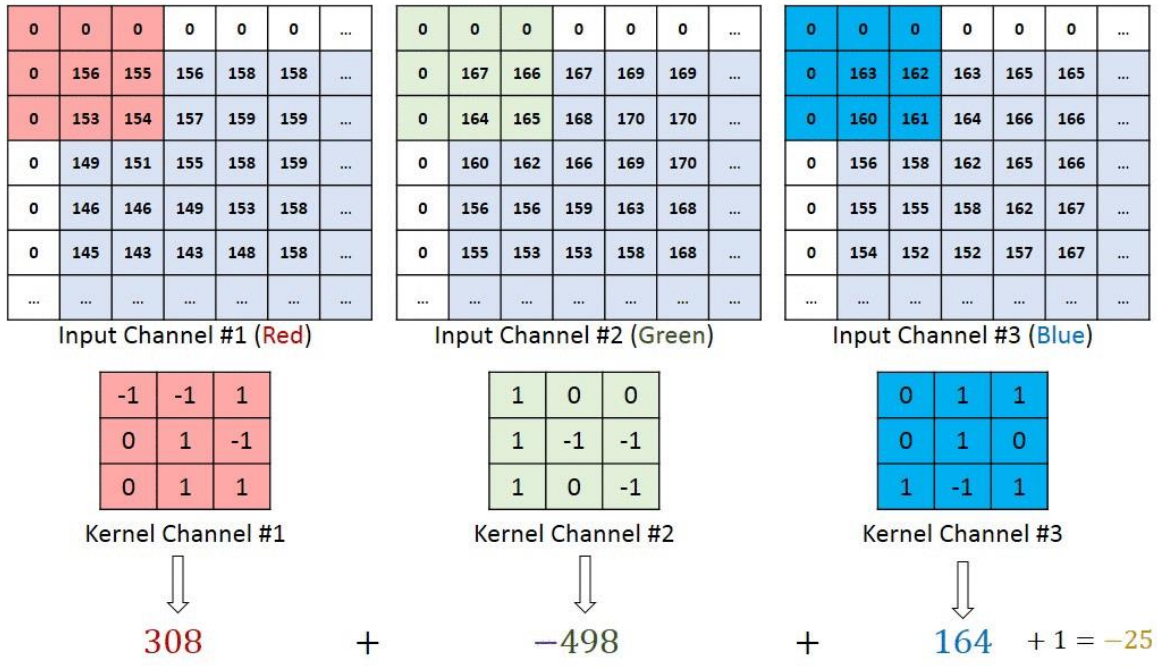


Figure modified and taken from [9]

1. We operate on images, particularly the **RGB** channels
2. The **convolutional layers** act as **encoders** for feature representations
3. From those representations we obtain predictions
4. We compare predictions with the **ground truth (gt)** via a loss function
5. Backpropagate the loss using gradient descent
6. Update our weights, i.e. **kernel entries**

Convolution operation in CNNs

- Invented in 1979 [20] – see [8] for the Stanford course
- We learn kernel entries using backpropagation



Animation graciously taken from [21]

Discrete convolution

$$(f \star h)[n] = \sum_{m=-M}^{M} f[n-m]h[m] \quad (13)$$

Easily differentiable

$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x} h) \star f \quad (14)$$

★is typically used for the convolution (sum-of-products) operation

Typical CNN architecture

- Convolutional layers, activations, pooling etc. → feature-maps
- Fully connected (FC) layer followed by softmax
- Obtain a score (0, 1] at every neuron → maximum-a-posteriori rule for classification
- Differentiable cross-entropy loss → we can use gradient descent

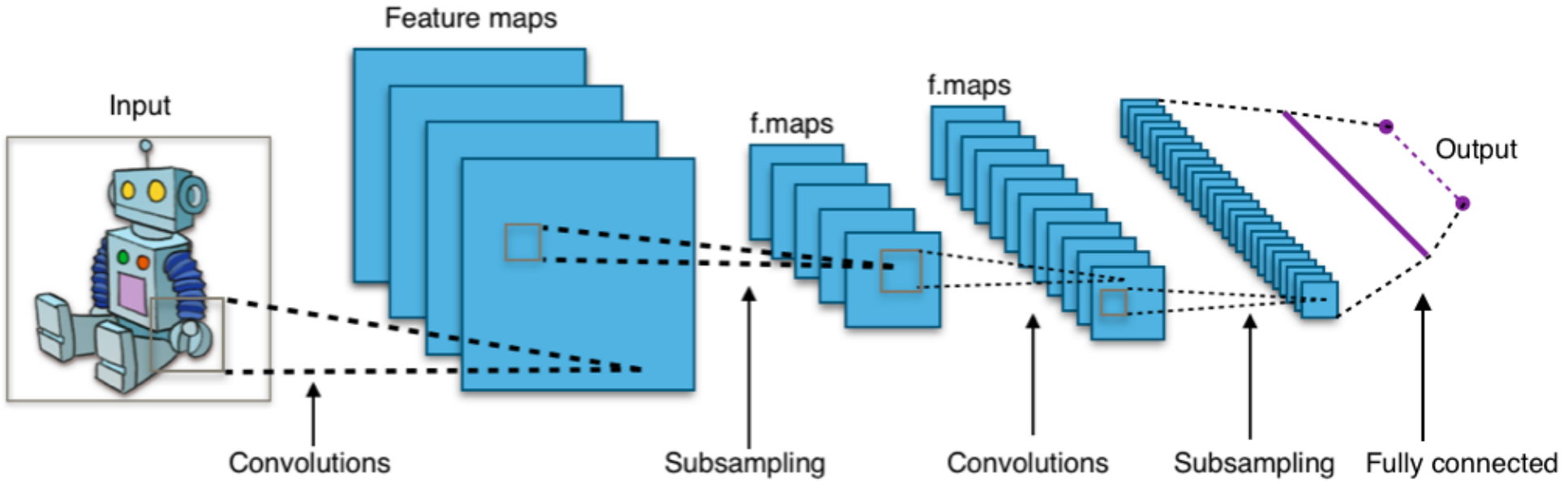


Figure taken from [22]

CNNs as powerful feature extractors

- Augmented RGB images
- The features become increasingly complex
- Visualizations on the right [23] are „projected activations of selected feature maps“
- No heatmaps but highlights of contributing features

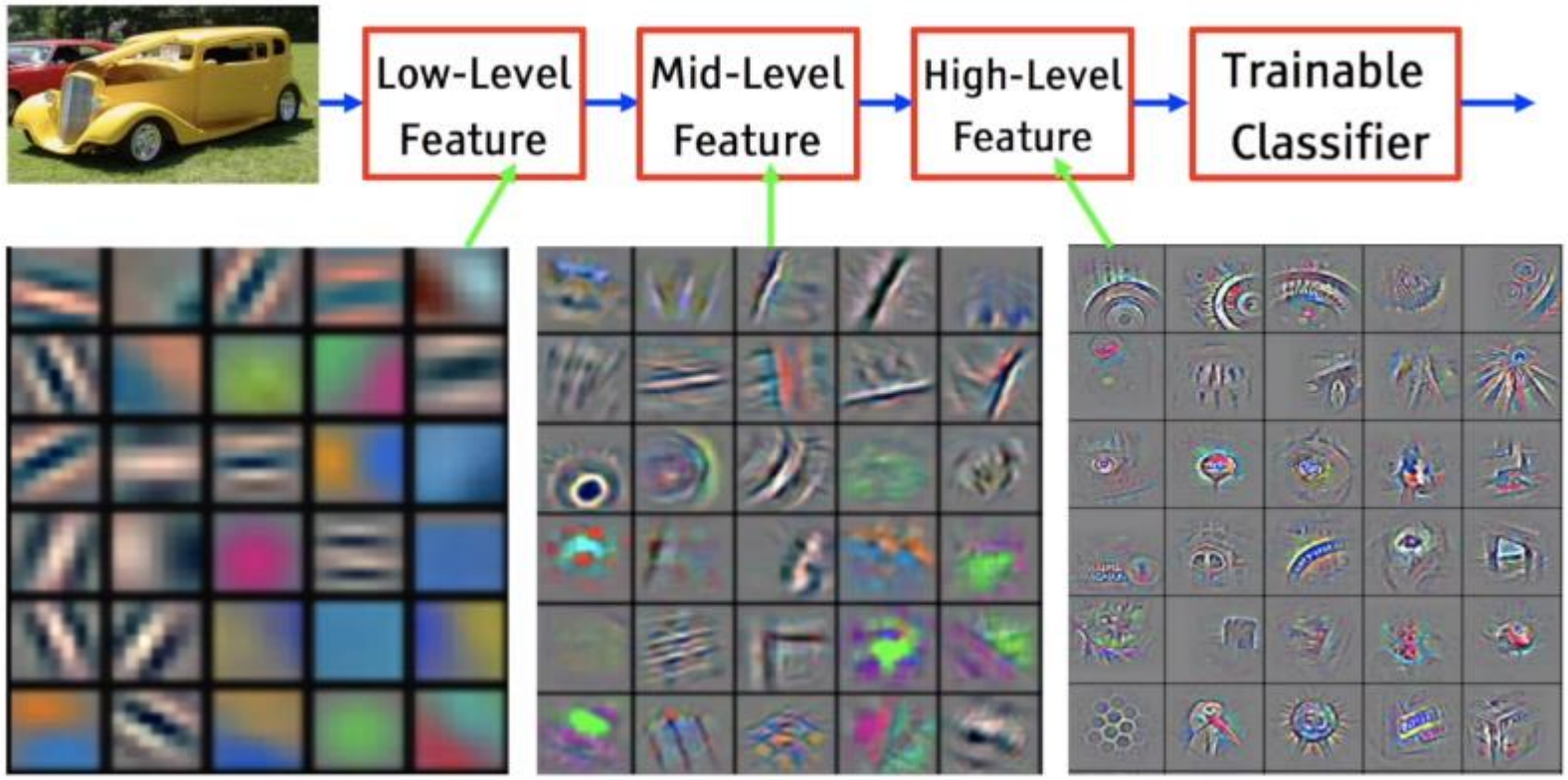


Figure taken from [23]

Gradient-Class Activation Maps (Grad-CAM) [24]

- Uses gradient information flowing into last convolutional layer
- Assigns specific neurons more/less importance
- Red regions correspond to „high scores“ for that class. The method „Guided Backprop“ was proposed by [25]

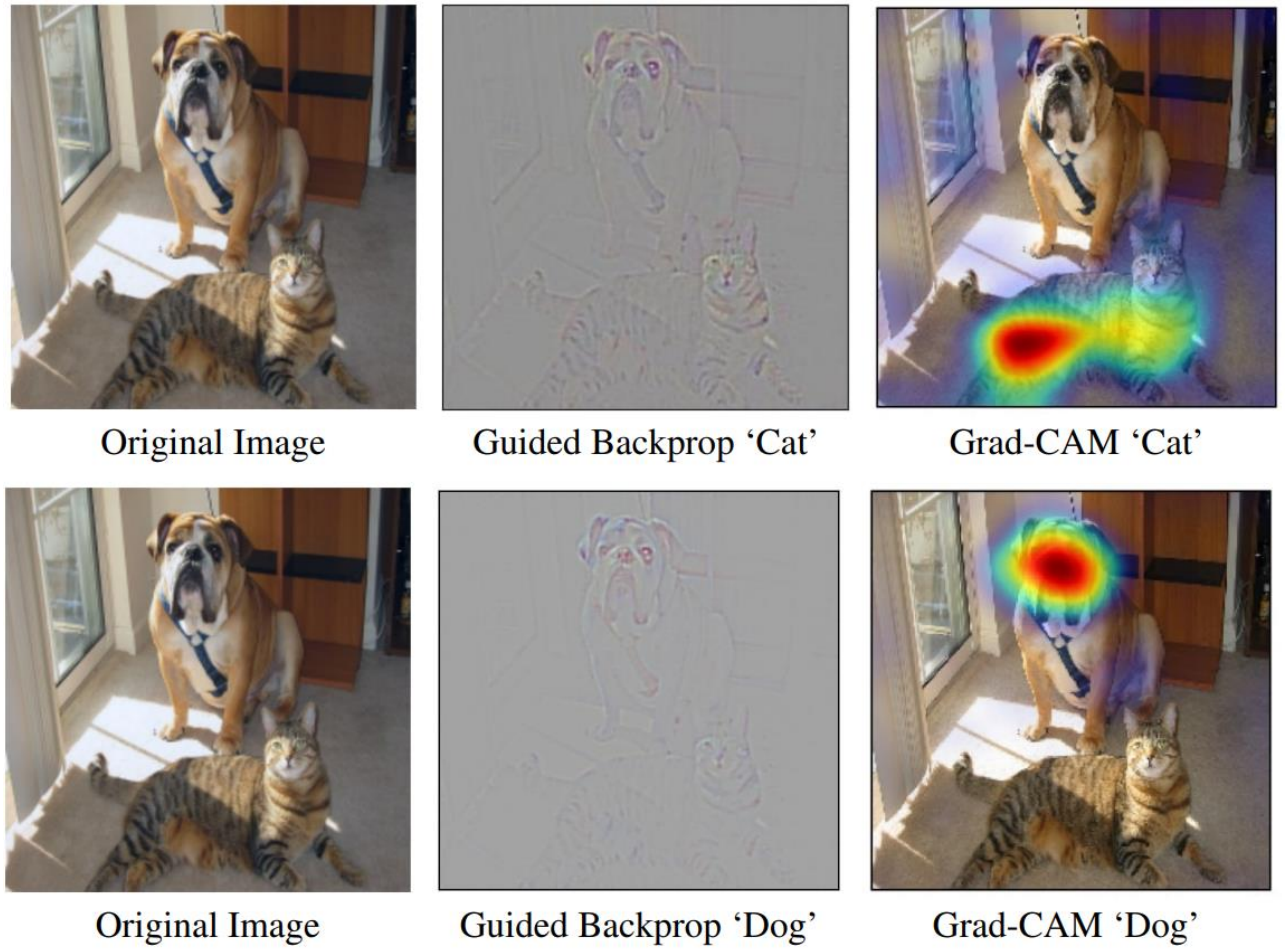
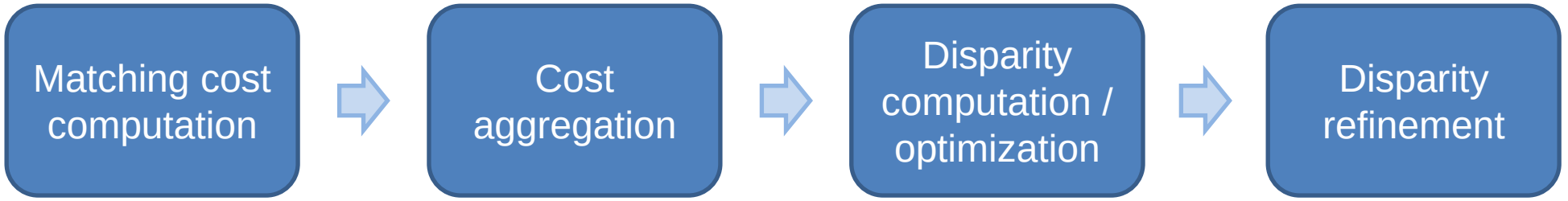


Figure slightly modified and taken from [24]

4 steps of stereo vision



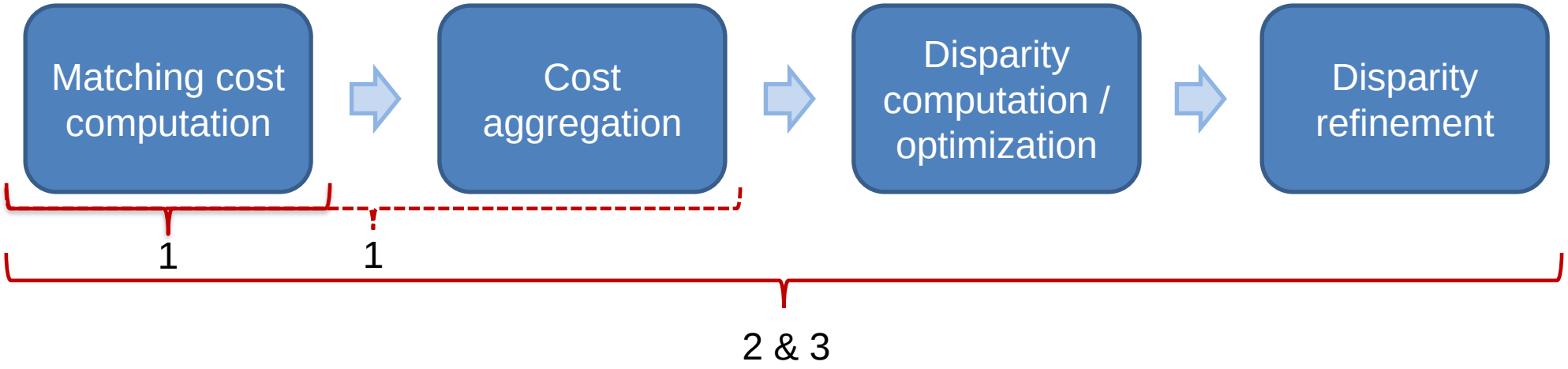
4-step stereo vision methodology from [30] – figure inspiration from [31]

1. Matching cost as difference between pixel intensity (classical) or image patches (CNNs)
2. Cost aggregation via window-based summation, global cost, cross-based or filters
3. Winner-takes-all, i.e. lowest cost for all possible disparity map values
4. Slanted plane smoothing, left-right-consistency or gaussian filters for smoothing

- CNNs replace one or more of these components

CNNs as replacements

Figure inspiration from [31] – below 3-level taxonomy following [32]



1) **Non end-to-end**: MC-CNN [33], improved with multiscale-fusion [34] or multilabel distribution [35]
   Uses feature maps to construct matching costs using e.g. dot product or fully-connected -> sigmoid

2) **End-to-end**: all steps jointly optimized, using a) **2D encoder-decoder** [36] or b) **3D convolutions** [37]
   Methods for optical flow and stereo matching but stereo search space limited due to epipolar constraint

3) **Unsupervised**: rely on minimizing photometric warp error – difference between true and warped image
   DeepStereo generates new views using nearby images [38] – [39] include pixel-wise reconstruction loss

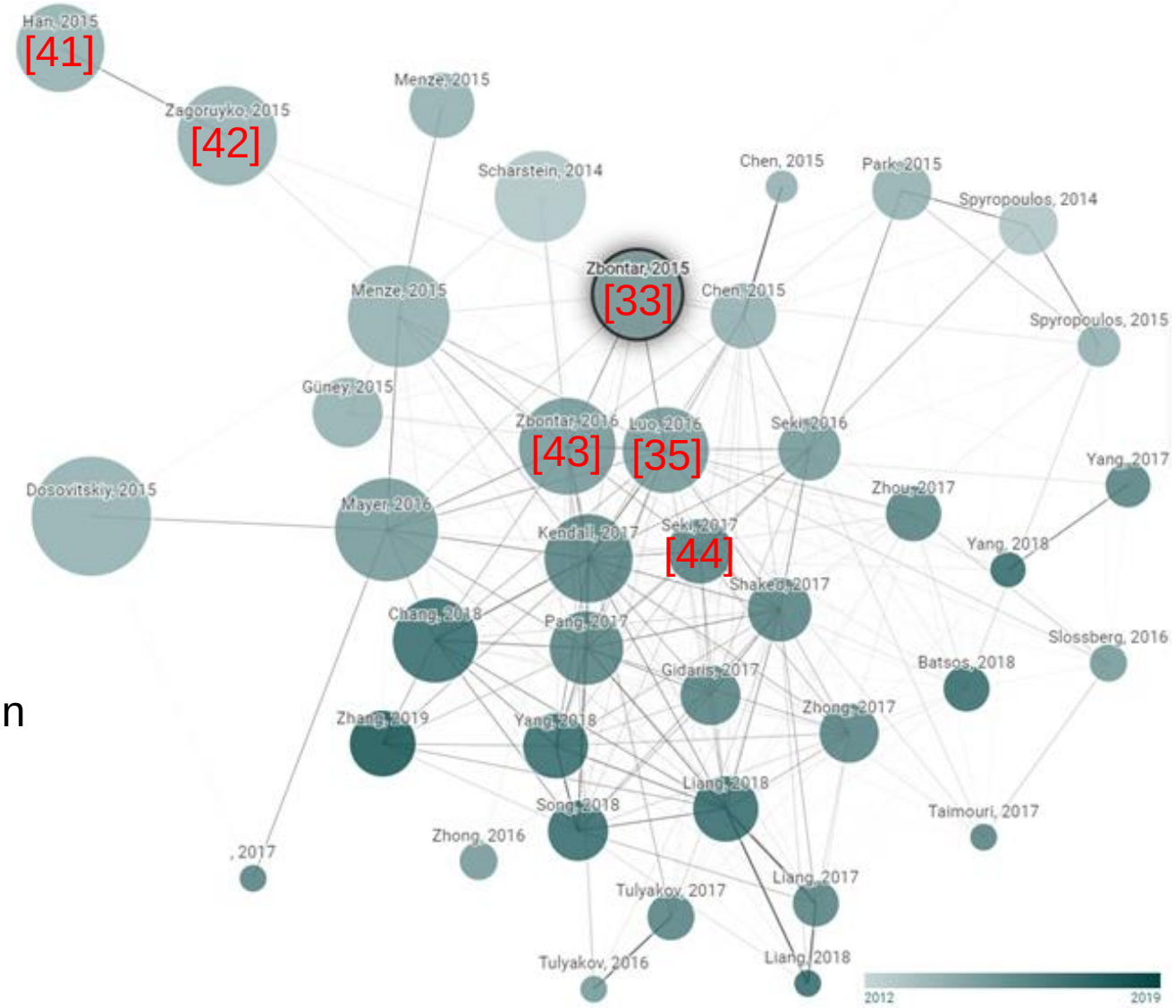
Non-end-to-end deep stereo matching methods

General similarity models

[41]: MatchNet – Han, 2015
[42]: ZagoruykoNet – Zagoruyko, 2015

Stereo matching and disparity estimation

[33]: MC-CNN – Zbontar, 2015
[43]: MC-CNN variants – Zbontar, 2016
[35]: Content-CNN – Luo, 2016
[44]: SGM-Net – Seki, 2017

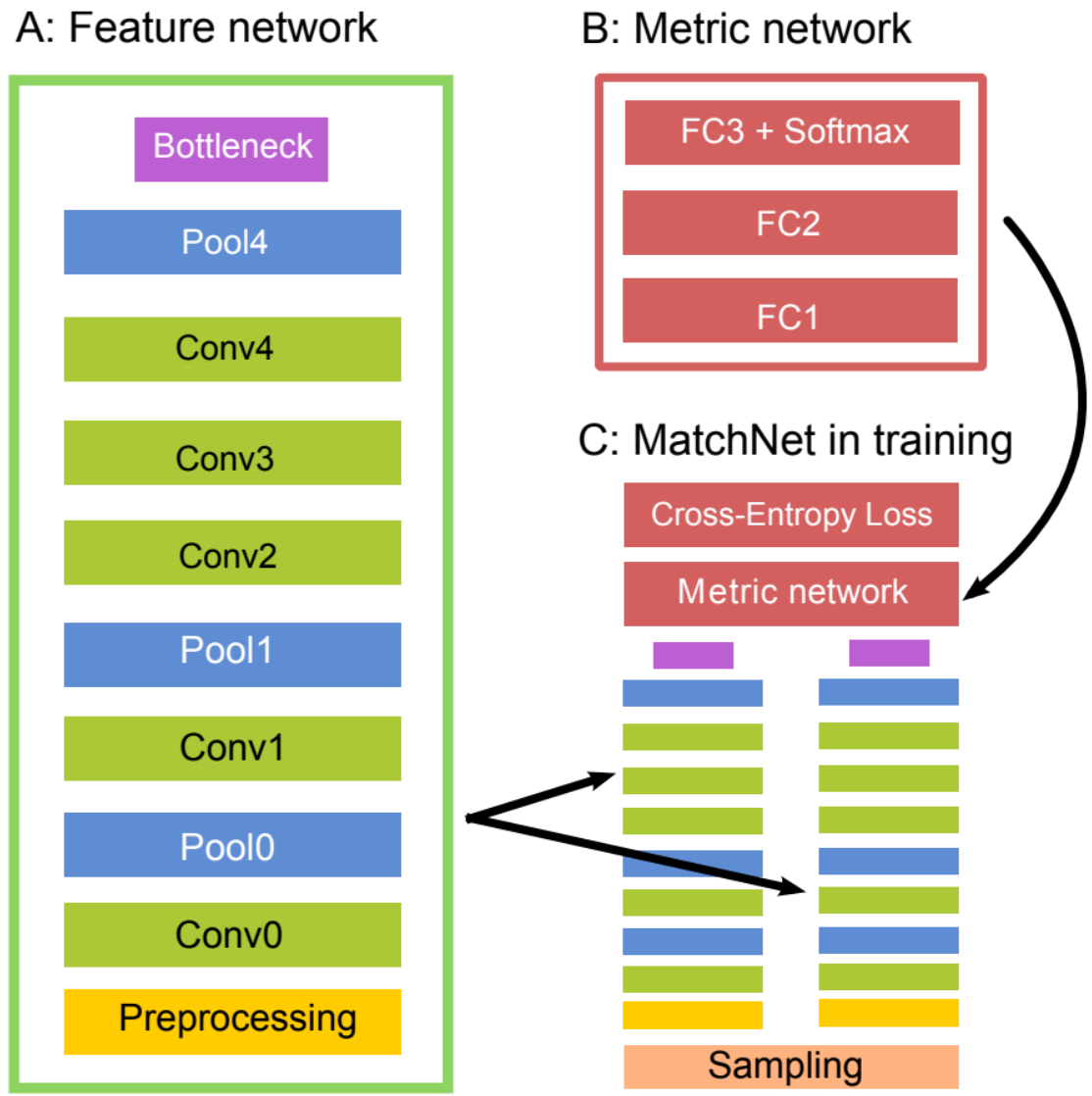


Slightly modified ConnectedPapers [40] graph for [33] as of May 2021

- **MatchNet** [41] - two AlexNet [15] models for feature extraction and a metric network (3 FC + softmax) on image-pairs

- Mimize cross-entropy with y_i as 0/1 label for input pair x_i with $y_i = 1$ for matches

$$E = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (15)$$



The architecture of MatchNet - figure taken from [41]

- **ZagoruykoNet** [42] - general similarity function for two image patches

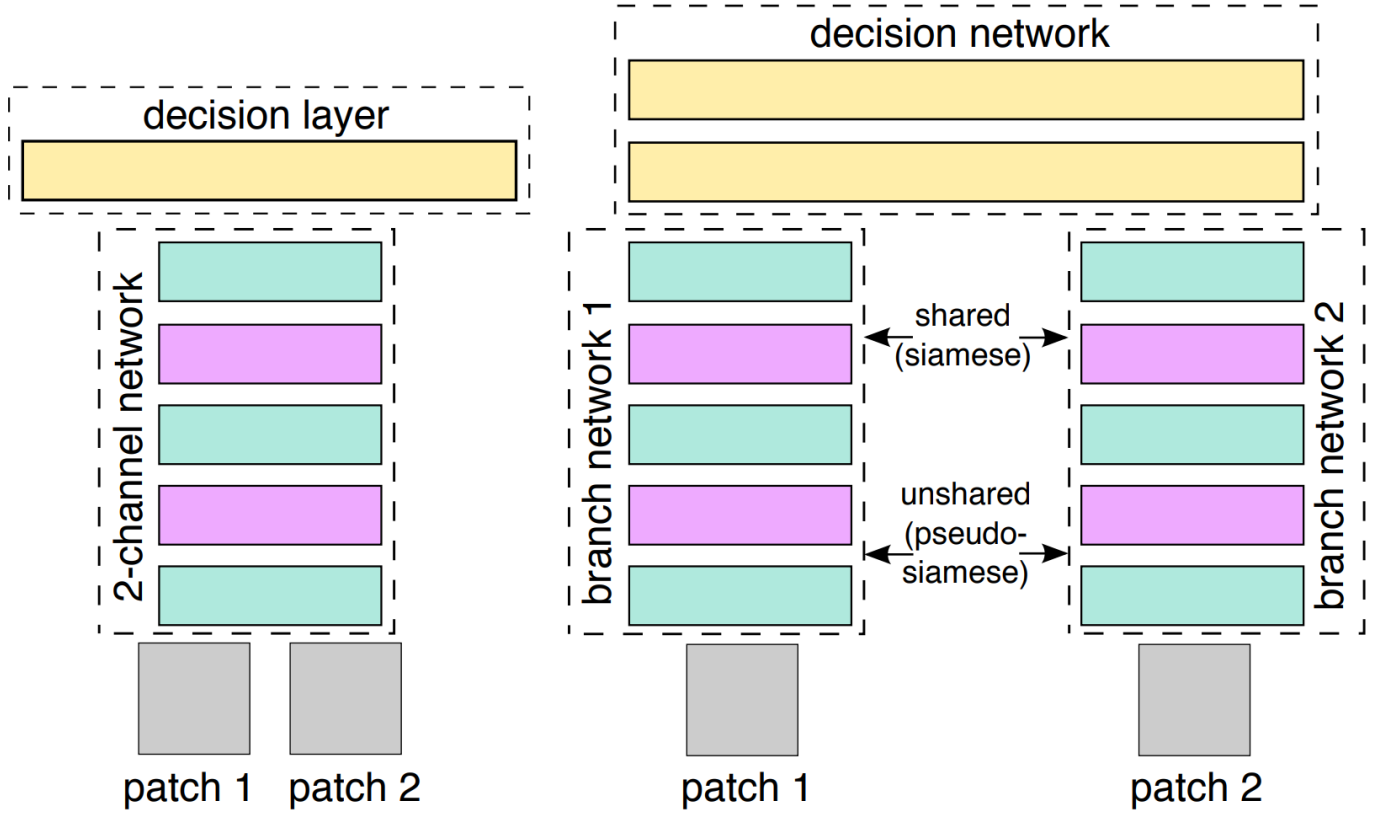
a) 2-channels: 2 patches of input pair as 2-channel image

b) Siamese: two branches with shared weights

c) Pseudo: two siamese branches with some unshared weights

$y_i \in \{-1, 1\}$ ... match-indicator

$$\min_w \frac{\lambda}{2} ||w||_2 + \sum_{i=1}^{N} \max(0, 1 - y_i \hat{y}_i) \quad (16)$$

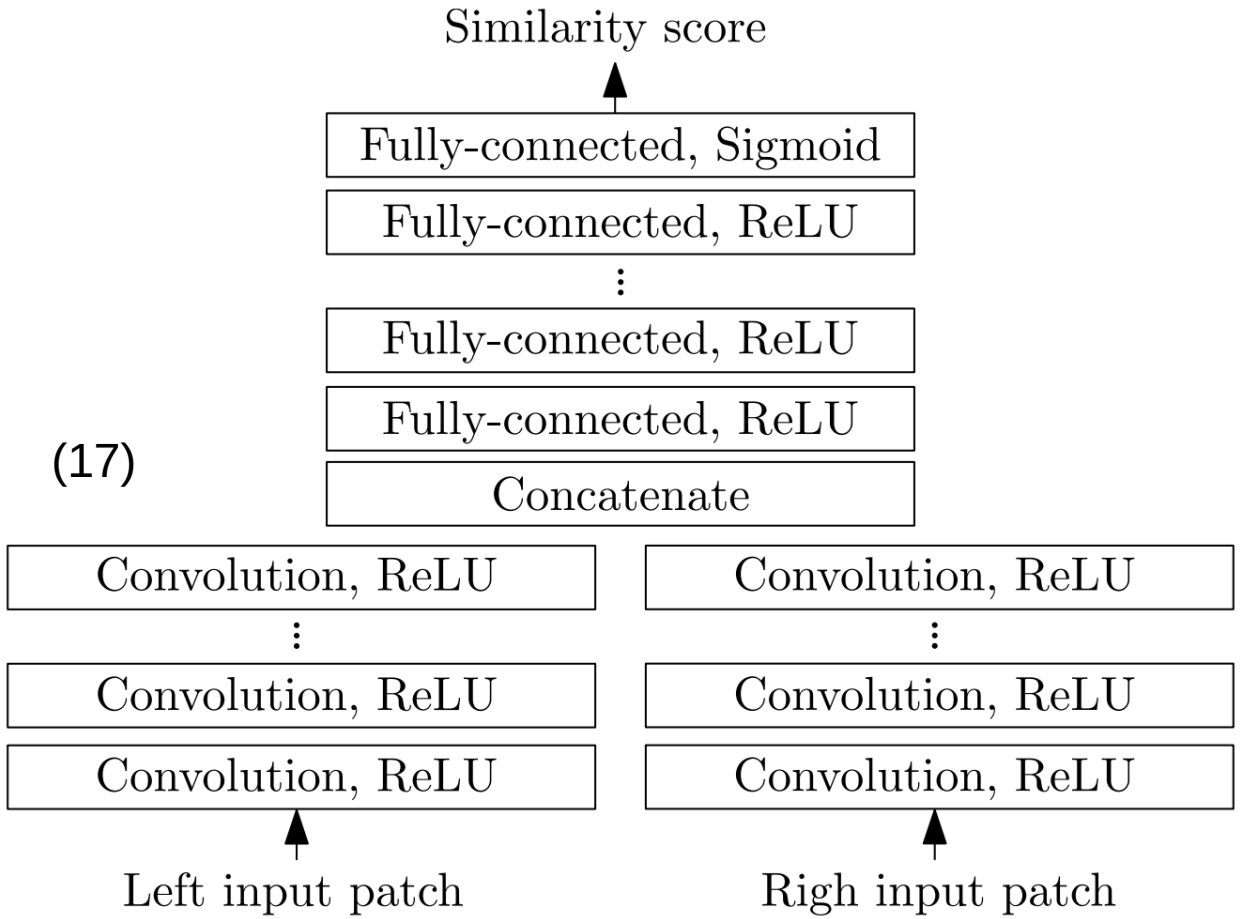


Three basic architectures a), b) and c) - figure taken from [42]

- **MC-CNN** [33, 43] - same architecture **specifically** for matching cost computation in stereo matching
- Matching cost via CNN, $\boldsymbol{p}$ as image positions and disparity d

$$C_{CNN}(\boldsymbol{p}, d) = f(< P^{L}_{9\times9}(\boldsymbol{p}), P^{R}_{9\times9}(\boldsymbol{p} - \boldsymbol{d}) >) \quad (17)$$

- Individual forward pass for every disparity value
- [43] also introduces a fast variation with a dot-product instead of FC-layers



Siamese architecture, „accurate variation“ – figure taken from [43]

- **Content-CNN** [35] - computes inner product between two siamese representations

- Further treat it as a multi-class classification problem, yielding distribution over disparity candidates

Check out the 3-minute presentation of Content-CNN [35] for the CVPR 2016

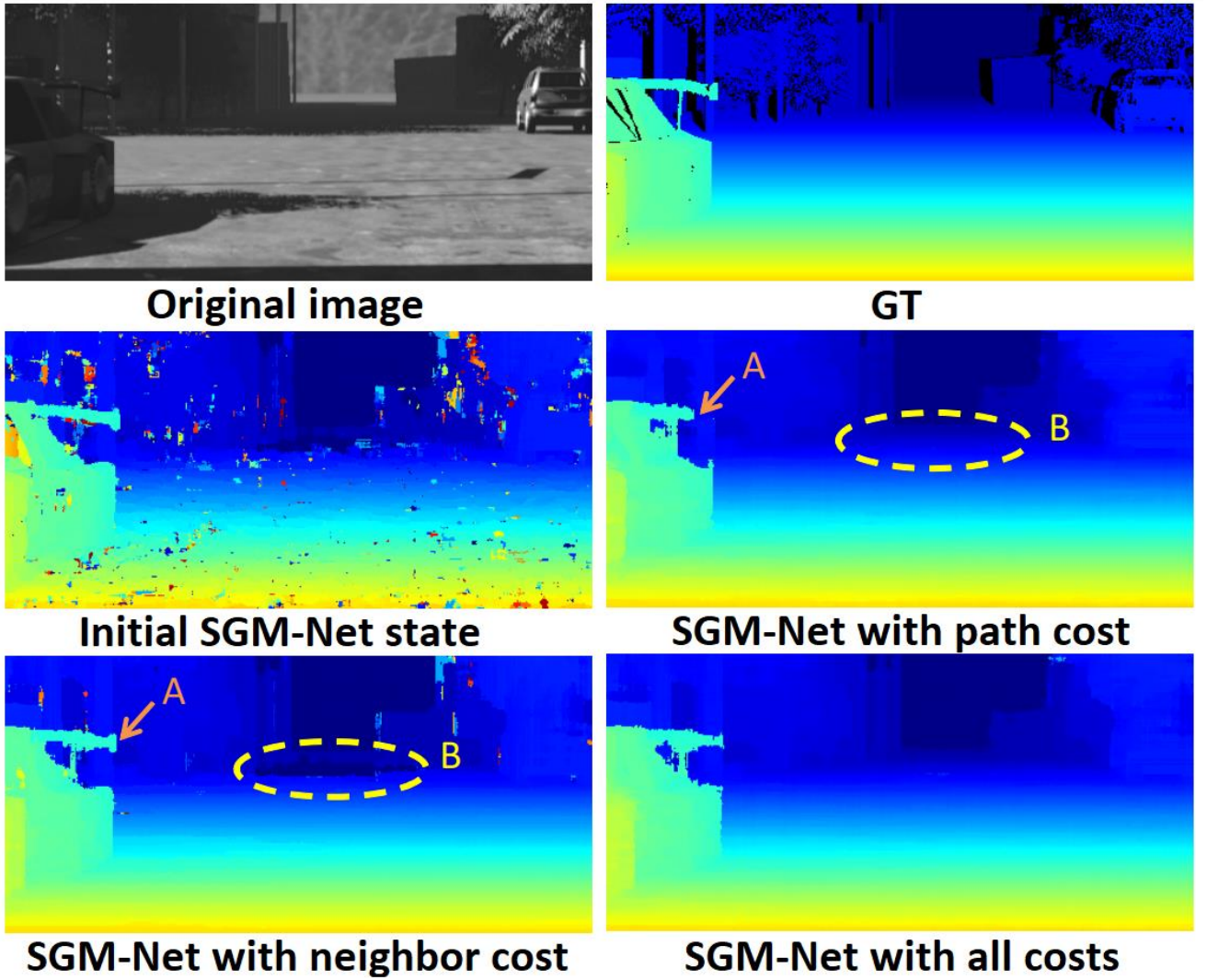
https://www.youtube.com/watch?v=EEqCf_eno5c

- **SGM-Net** [44] – regularization via semi-global matching (SGM) [57]

- CNN is used for matching and learning SGM penalty paramaters

a) path-cost: path traversing correct disparity at a pixel should have smaller cost than any other paths

b) neighbor-cost: path traversing correct disparities in consecutive pixels must have smallet cost – deals with borders, slants and flat regions



Qualitative comparison results for loss functions – figure modified and taken from [44]

End-to-end super-vised 2D CNNs for stereo vision

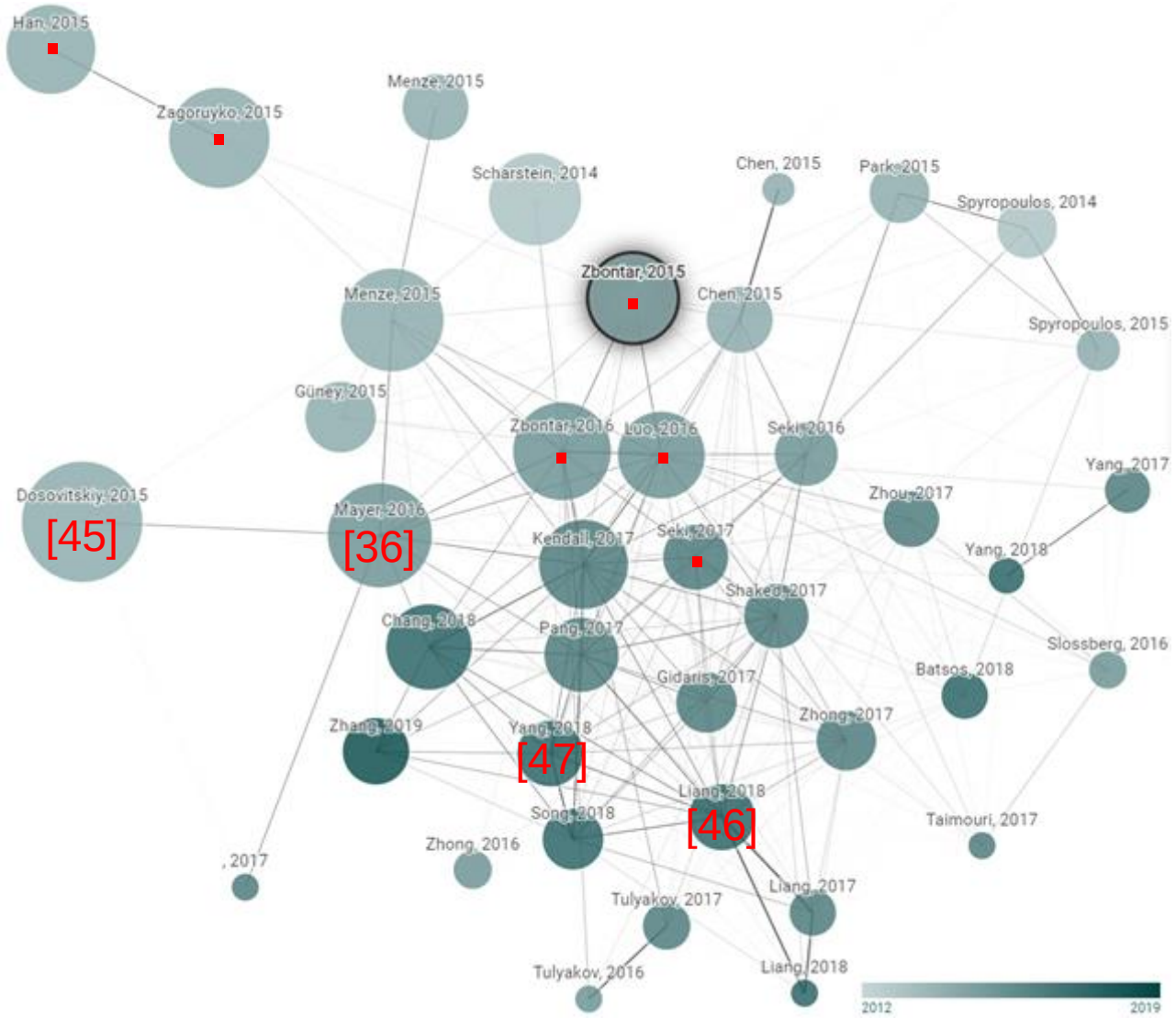
Optical flow works for stereo vision

[45]: FlowNet – Dosovitskiy, 2015

2D CNNs for entire stereo pipeline

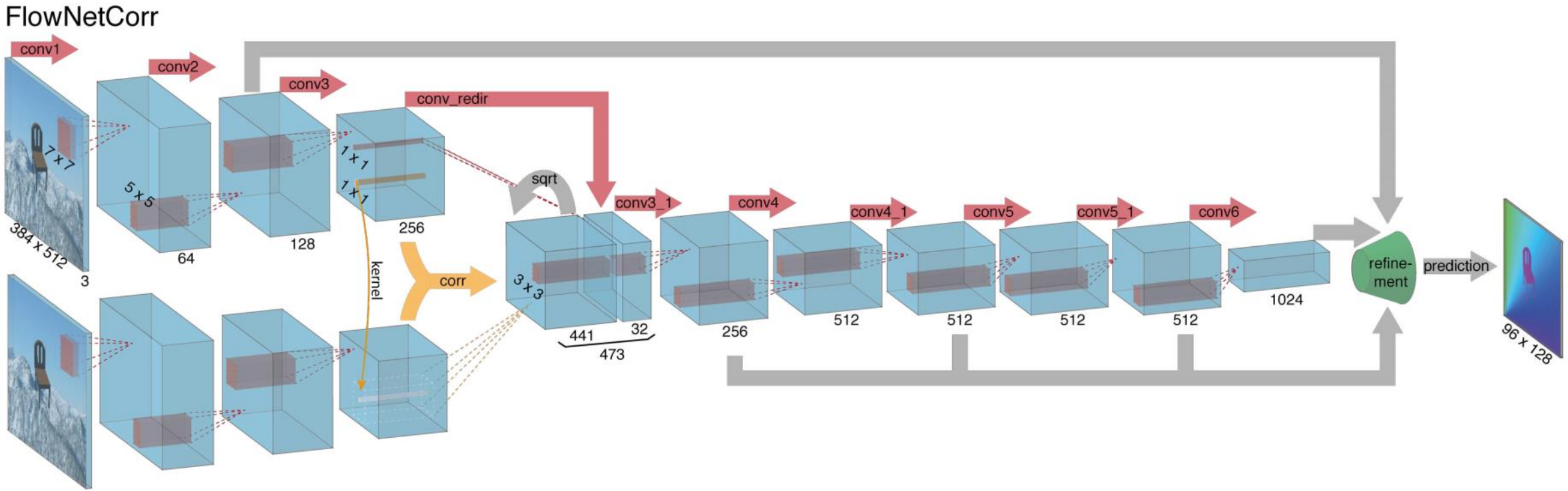
[36]: DispNet – Mayer, 2016
[46]: iResNet – Liang, 2018
[47]: SegStereo – Yang, 2018

[62] HITNet as current top-performer (Slide 45f., preprint for CVPR 2021)



Slightly modified ConnectedPapers [40] graph for [33] as of May 2021

- **Flownet [45]** – first end-to-end network for stereo matching, proposed for optical flow
- Provides basic 2D encoder-decoder, matching via „correlation“ of two patches: convolution but data is convolved with other data
- Decoding (upsampling) takes place in the refinement



Red steps are convolutions, in yellow the correlation is obtained, while green modules provide the refinement – figure taken from [45]

- **DispNet [36]** – inspired by FlowNet [45], propose dataset & networks for disparity estimation, optical flow and scene flow estimation
- First end-to-end CNN method explicitly for disparity estimation
- Architecture very similar to FlowNet with additional convolutional layers

- **iResNet [46]** – network in three parts
  1. Calculate multi-scale shared features from left/right images
  2. Perform cost computation, aggregation and disparity calculation
     - Calculate „feature constancy“ using initial disparity and features
  3. Disparity and feature constancy fed into subnetwork for refinement

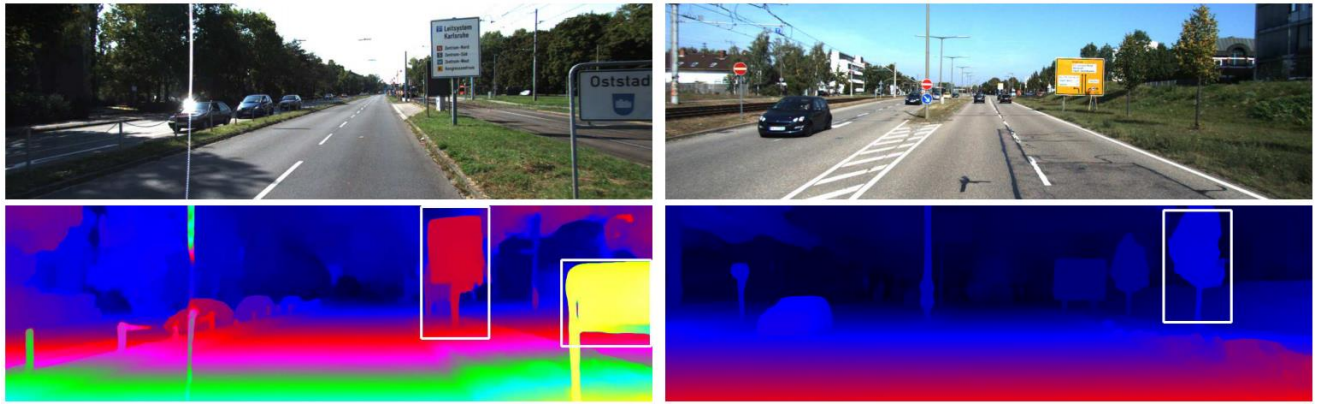
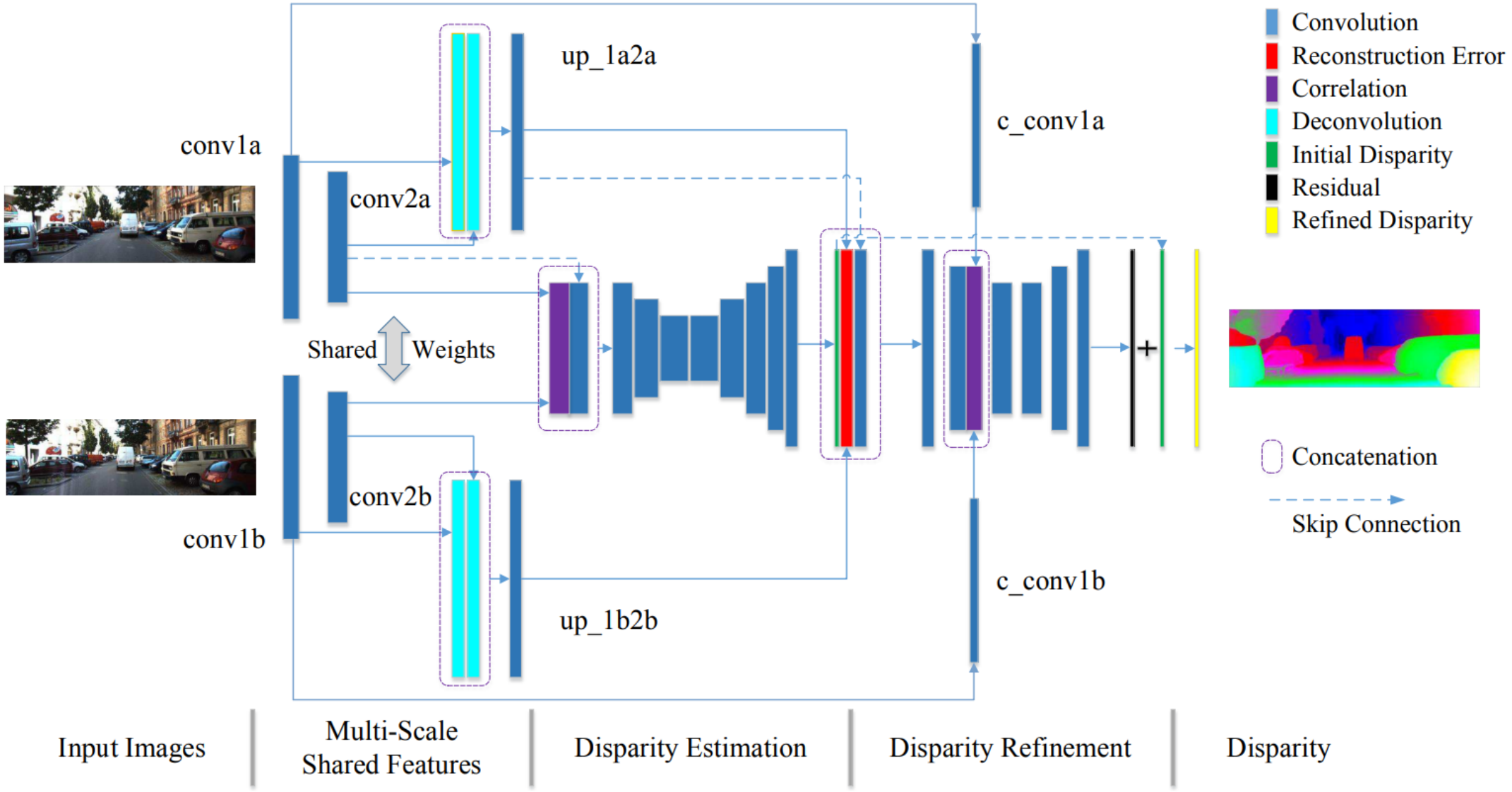


Figure taken from [46]
Top row: images from KITTI [48]
Bottom row: iResNet [46] disparity prediction

- **iResNet [46]** – network in three parts

Architecture of the iResNet network – figure taken from [46]



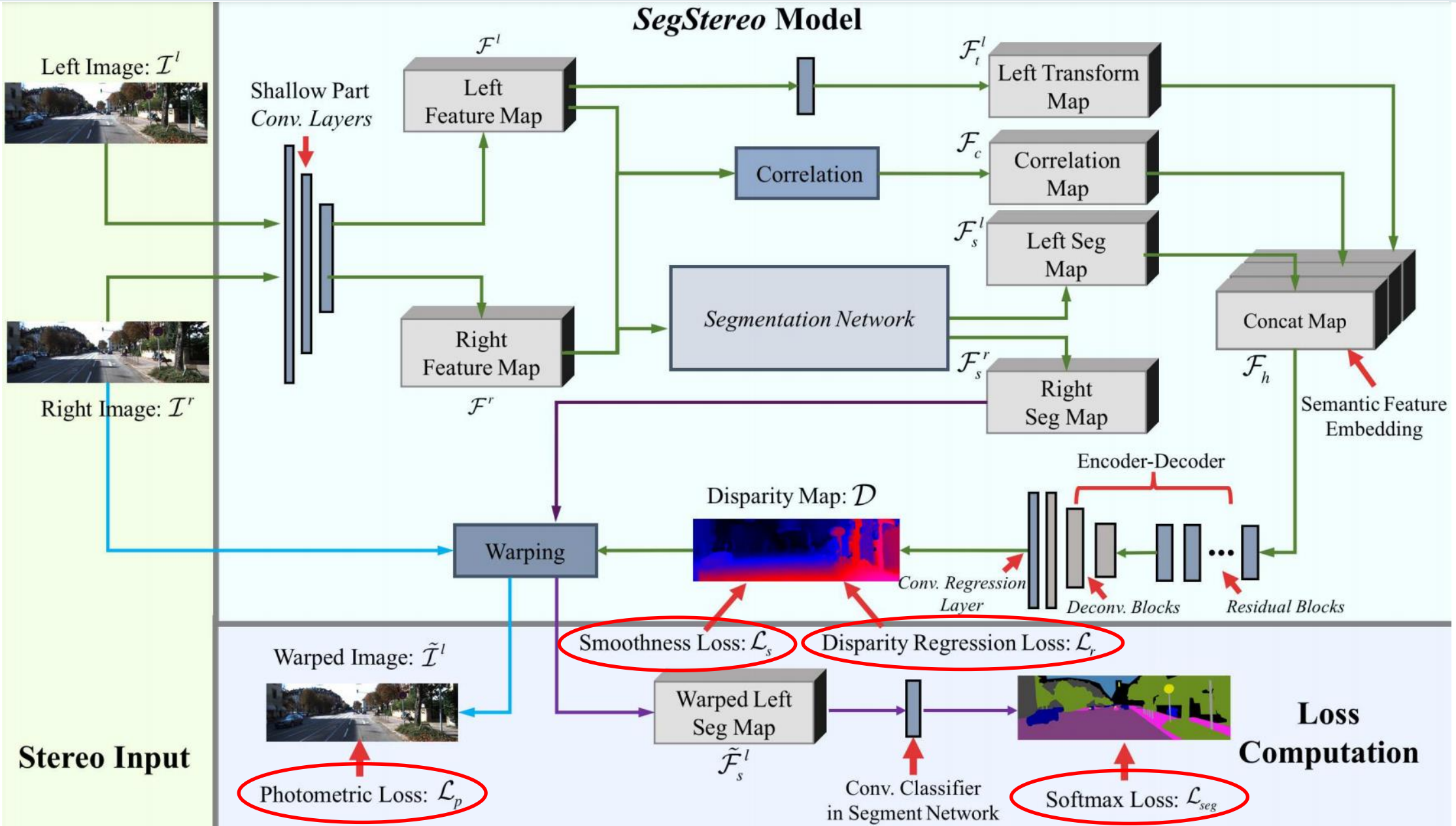


Figure taken from and architectural overview of **SegStereo [47]**

- **SegStereo [47]** argue that **semantic context** helps featureless regions
- Segmentation loss L_{seg} calculated from segmentation map and gt
- **Unsupervised: photometric reprojection loss** for view generation [49, 50]

$$L_p = \frac{1}{N}\sum_{i,j} \delta_{i,j}^{p} ||\tilde{I}_{i,j}^{l} - I_{i,j}^{l}||_1 \quad (18)$$

where $I_{i,j}^{l}$ and $\tilde{I}_{i,j}^{l}$ are the left image and warped right image respectively,$\delta_{i,j}^{p}$ is a mask for image borders & occlusion and N is the total number of pixels, each at position (i, j)

- With no regularization it can be locally incoherent – smoothness loss L_s

$$L_{unsup} = \lambda_p L_p + \lambda_s L_s + \lambda_{seg} L_{seg} \quad (19)$$

- Supervised: gt dispariy map $\widehat{D}$ for disparity regression loss, ν ... set of valid disparity pixels

$$L_r = \frac{1}{N_\nu}\sum_{i,j\in\nu} ||D_{i,j} - \widehat{D_{i,j}}||_1 \quad (20)$$

$$L_{sup} = \lambda_r L_r + \lambda_s L_s + \lambda_{seg} L_{seg} \quad (21)$$

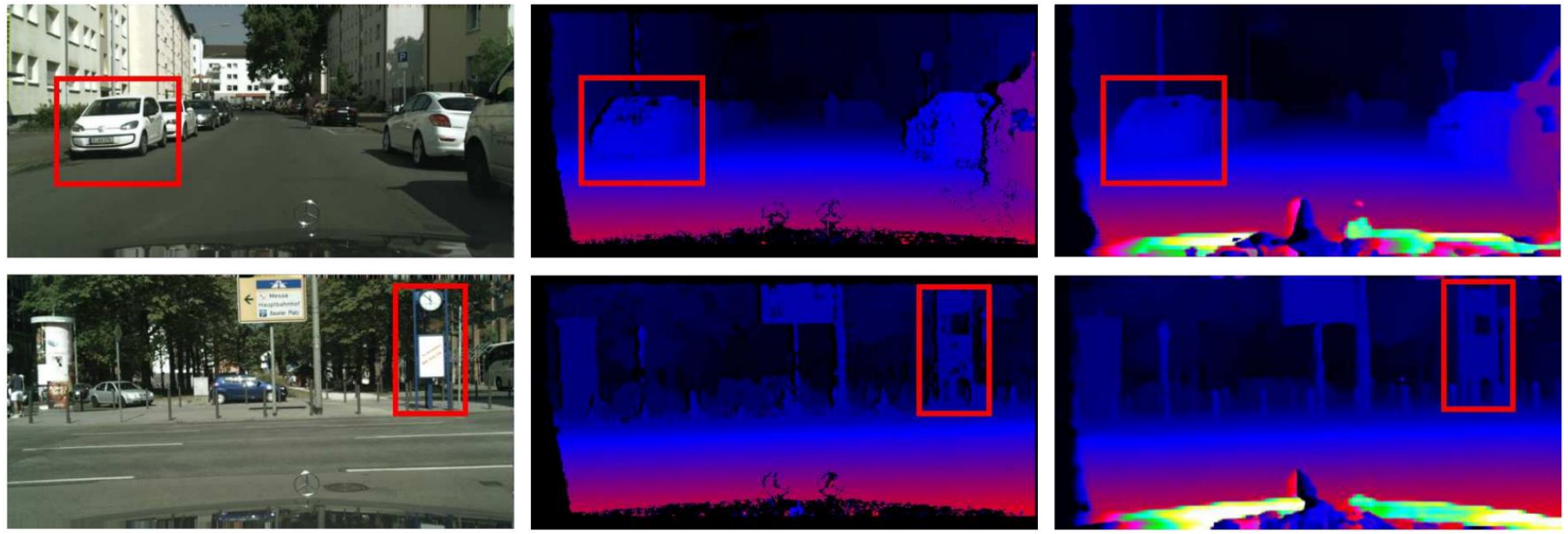


Figure taken from [47]

Left: input images from CityScapes [51]

Middle: SGMNet [44] results

Right: disparity results from an **unsupervised** SegStereo [47]

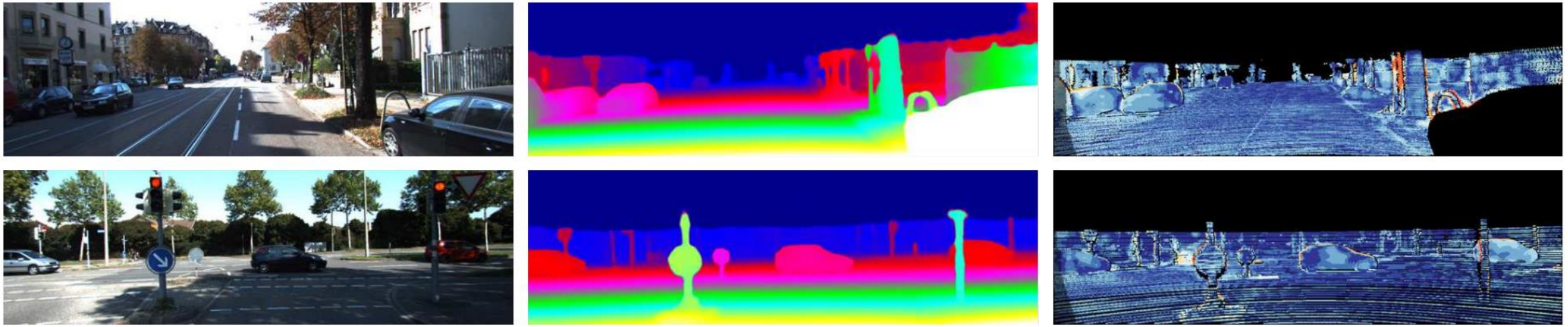


Figure taken from and results (middle) of a **supervised** SegStereo model [47] on KITTI [48] images (left) – error maps (right)

End-to-end super-vised 3D CNNs

3D convolutions to construct
4D cost volume

[33]: GCNet – Kendall, 2017
[52]: StereoNet – Khamis, 2018
[53]: PSMNet – Chang, 2018

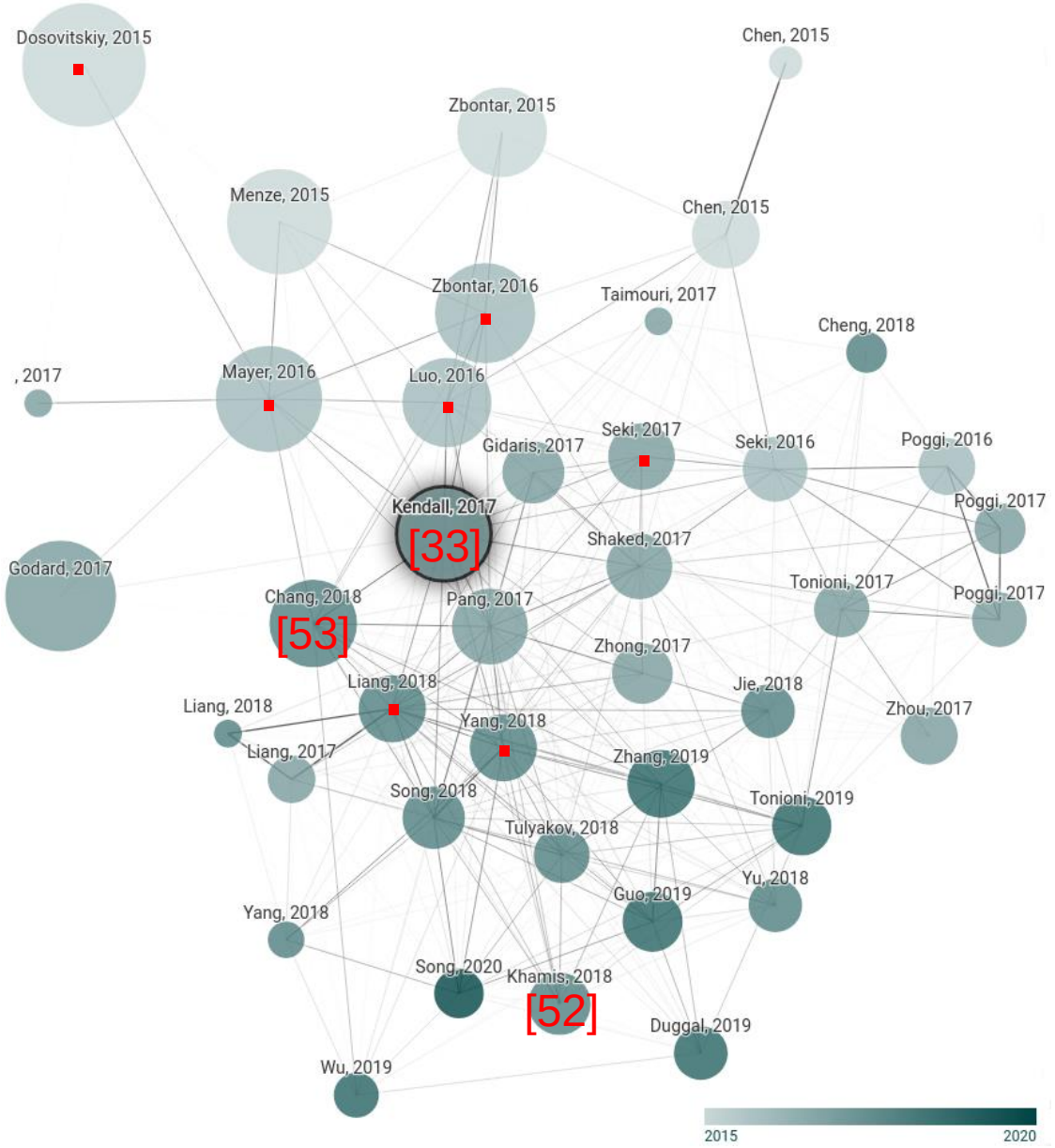


Figure on the right: slightly modified ConnectedPapers [40] graph for [33] as of May 2021

- **GC-Net [33]** – first to propose 3D convolutions

1. 2D convolutions create regular featuremaps with shared weights (siamese)
2. Each feature map is concatenated with all featuremaps across all disparities for the other image to form a „4D“ cost volume (see next slide)
3. 3D convolution then learns to regularize for width W, height H and disparity D
4. 3D deconvolutions yields final cost volume $\mathrm{H} \times W \times D$
5. Soft ArgMin, fully differentiable, regresses to disparities

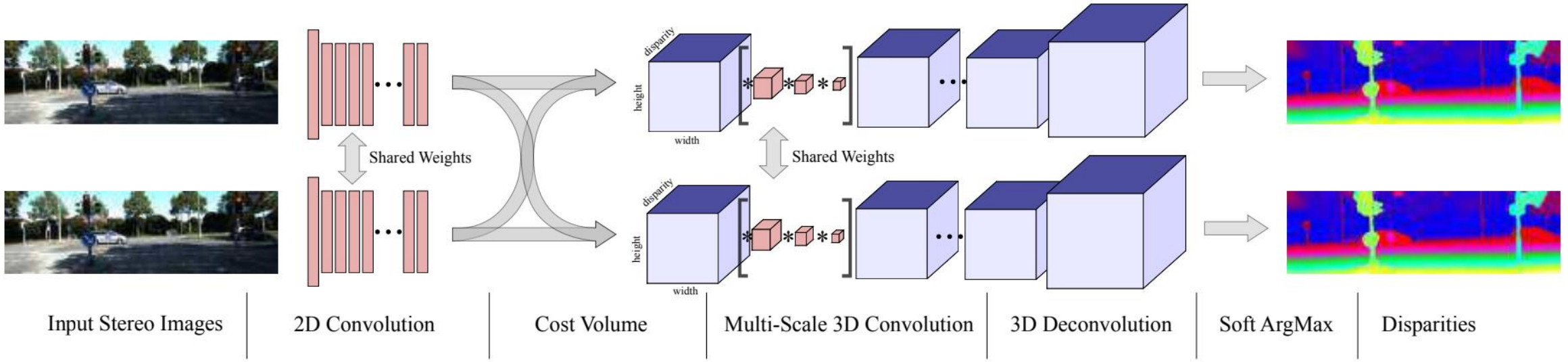
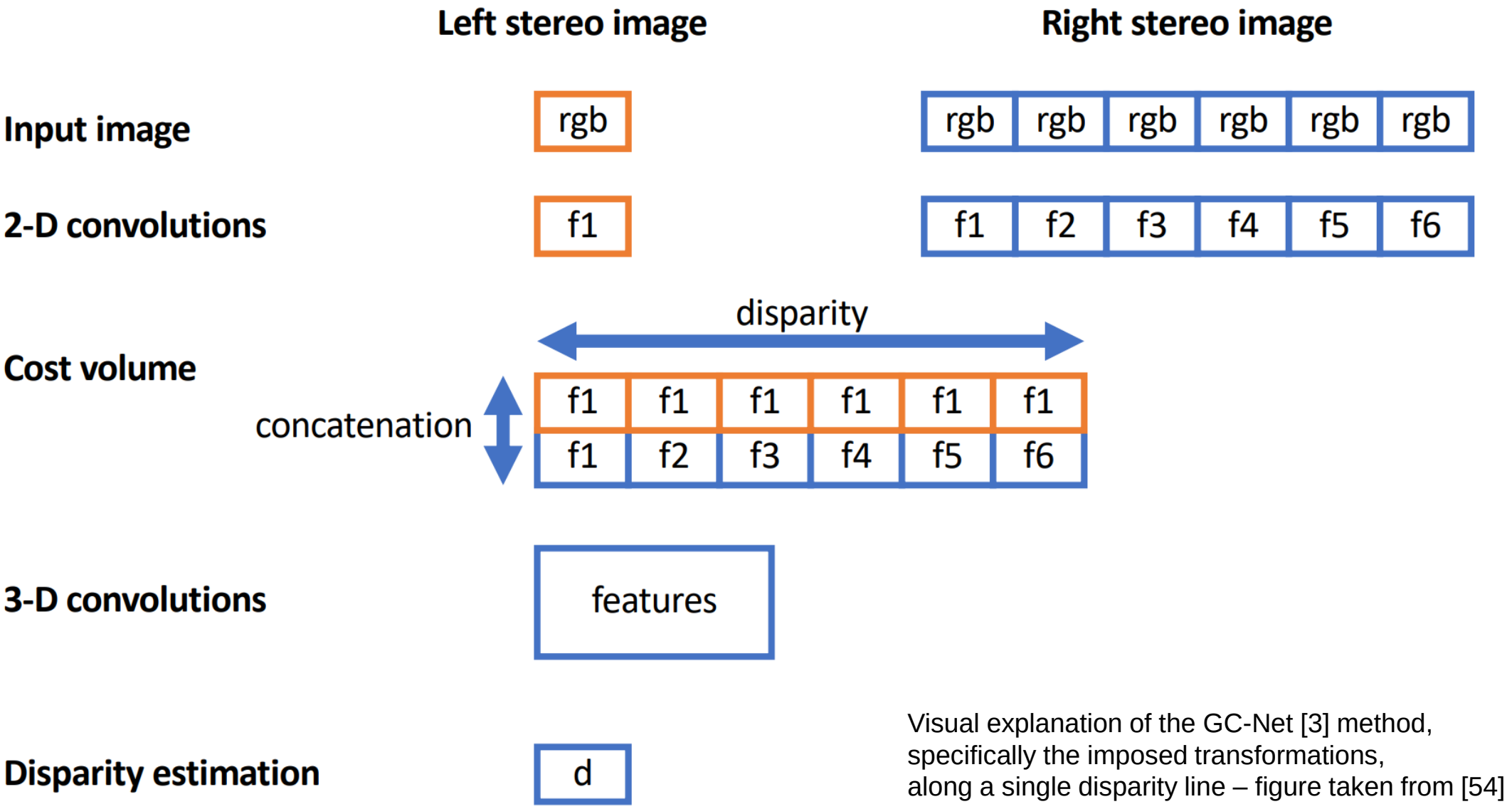
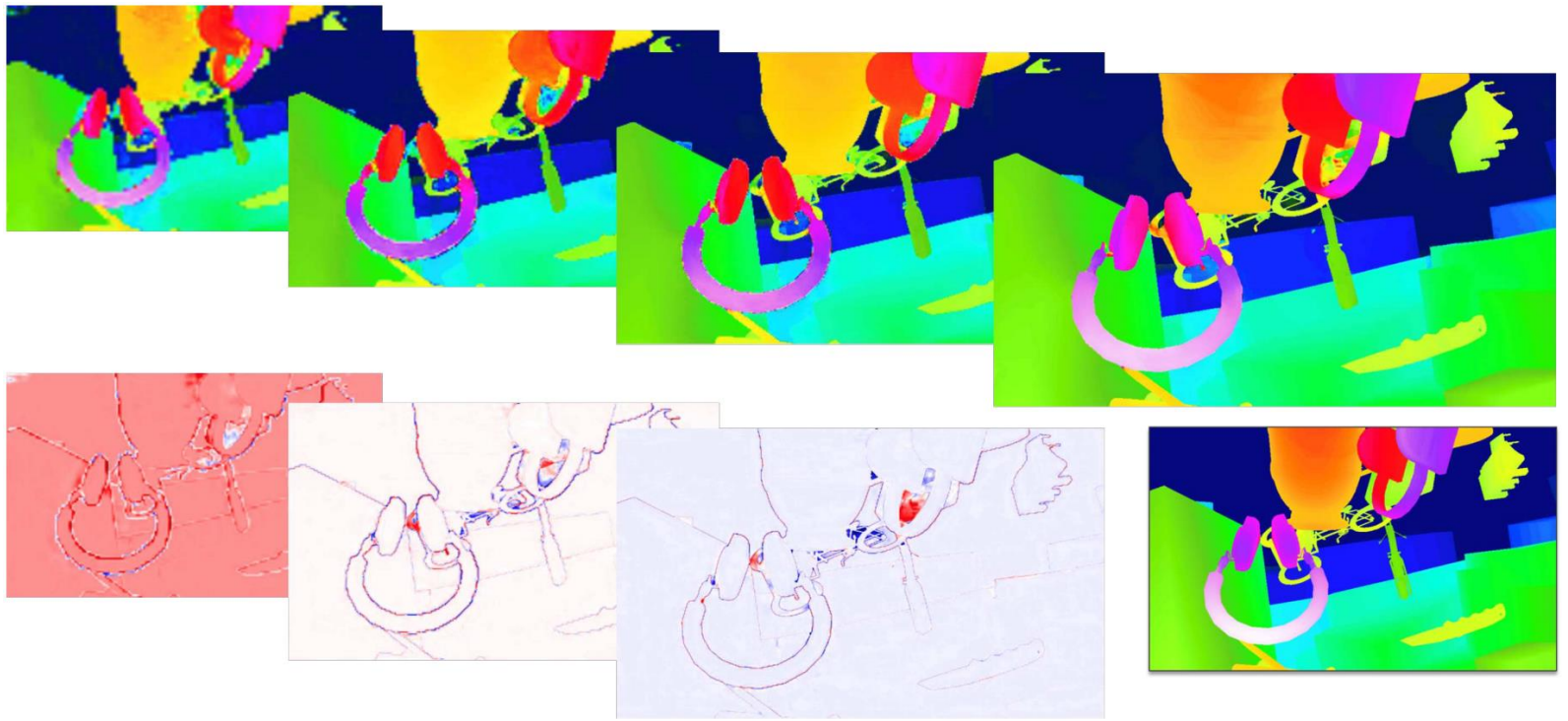


Figure taken from and architecture of GC-Net [33]



Visual explanation of the GC-Net [3] method, specifically the imposed transformations, along a single disparity line – figure taken from [54]

- **StereoNet [52]** – first end-to-end stereo matching algorithm for **real-time** (60fps, 1xTitan X)
- Subsample the features to $\frac{W}{2^k} \times \frac{H}{2^k} \times \frac{(D+1)}{2^k}$ with k downsampling layers
- Also use 3D convolutions followed by soft argmin but disparities are coarse
- For refinement disparities are dilated/eroded via a network that learns pixel-to-pixel mapping
- Intuition that network learns an **edge-aware upsampling function** with a RGB image guide



Hierarchical refinement steps and figure taken from StereoNet [52]

Top row gives the results and bottom row the refinement output at each stage

Ground truth in the bottom right

- **PSMNet [53]** – pyramid pooling for global context
- Similar to GC-Net [33] with a spatial pyramid pooling (SPP) module
- SPP concatenates representations from sub-regions with different sizes
- Stacked convolution – deconvolution pairs

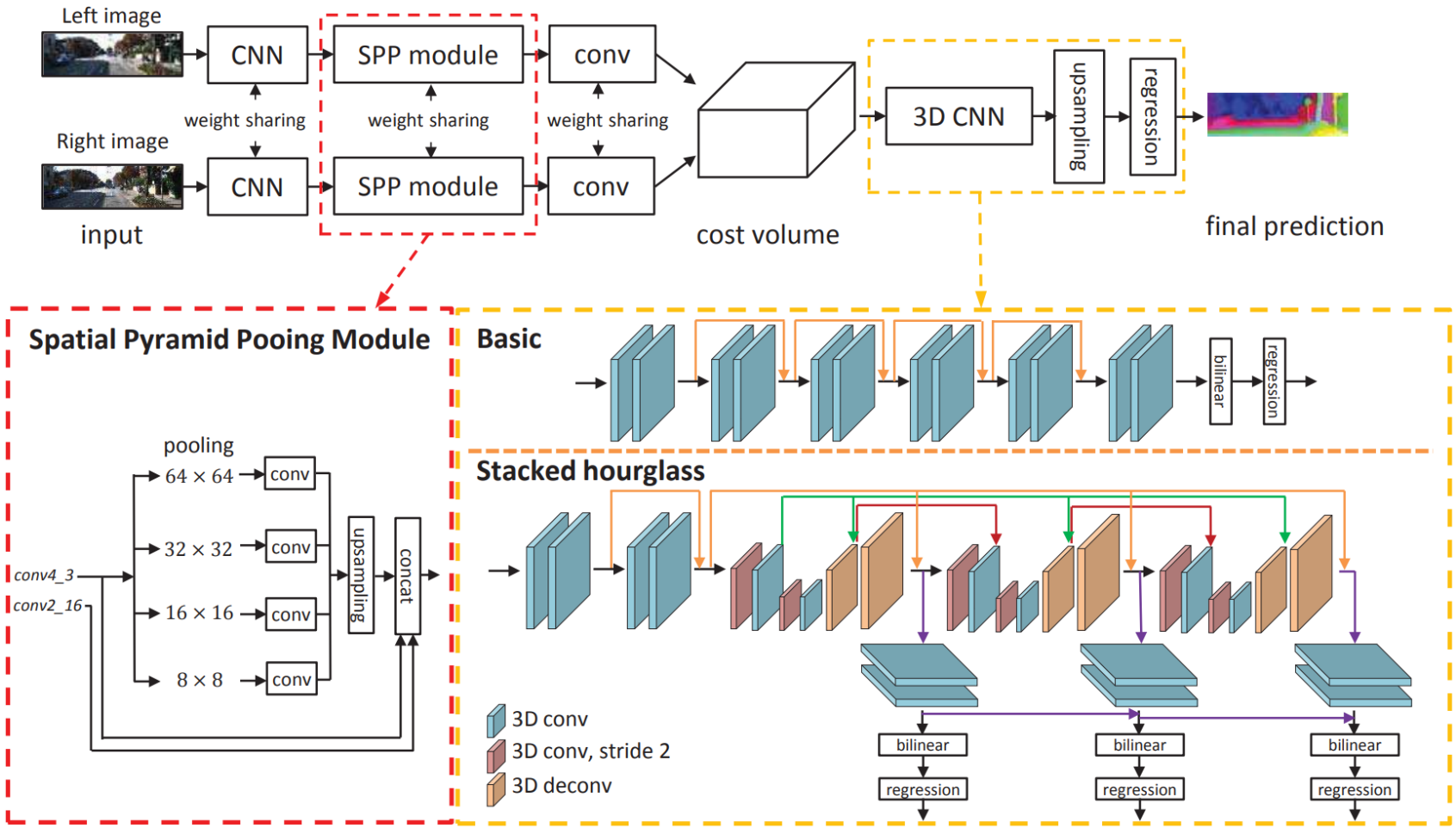
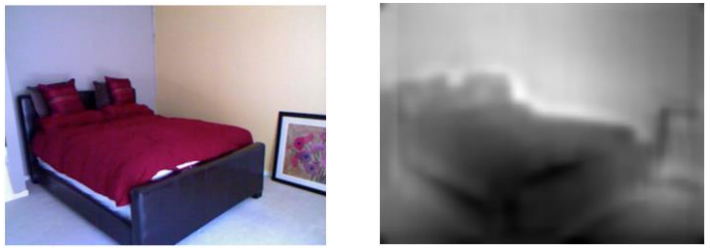


Figure taken from and architecture of PSMNet [53] – the SPP module and stacking the hourglasses are novel (in stereo vision)

Unsupervised Mono-Depth

First depth maps from single image in 2014 [55] - comparatively poor results



Figures taken from [55]

Photometric reprojection error [49] as work horse for unsupervised (monocular) depth estimation

[49]: GargNet – Garg, 2016
[50, 56]: Monodepth1/2 - Godard, 2017/9

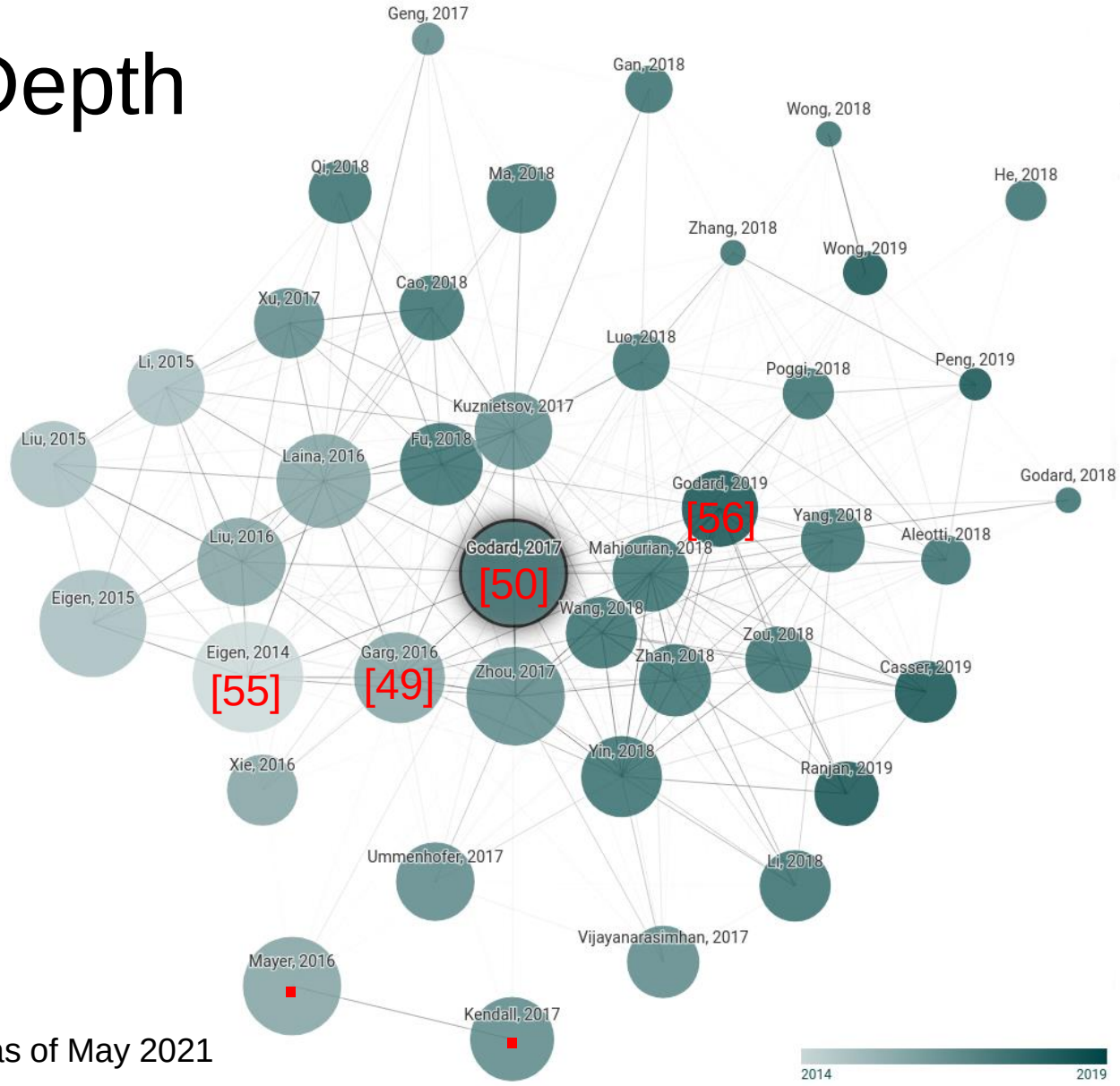


Figure on the right: slightly modified ConnectedPapers [40] graph for [50] as of May 2021

- **GargNet [49]** – unsupervised depth estimation, left image as source and right as target

1. Train encoder to predict depth map for source image $I^l_{i,j}$
2. Inverse warp the target image with depth and baseline to $\tilde{I}^l_{i,j}$
3. Optimize photometric error (see slide 32)

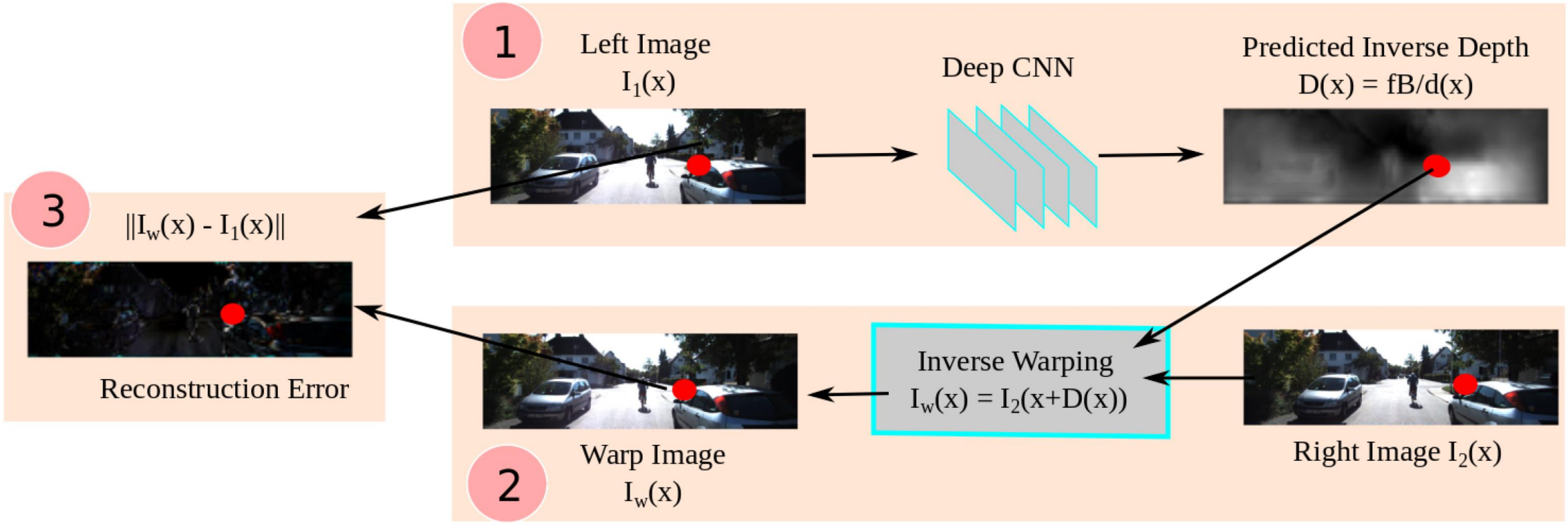


Figure taken from and architecture of GargNet [49]

- **GargNet [49]** – autoencoder (see slide 13) uses upsampling for last layers
- (-) Coarse and blurry predictions
- (+) 100% coverage since no pixels are occluded (compared to true stereo)
- SGM → CNN denotes a standard SGM algorithm [57] for proxy ground-truths to train a CNN
- HS → CNN similarly first uses the classical HS algorithm [58]

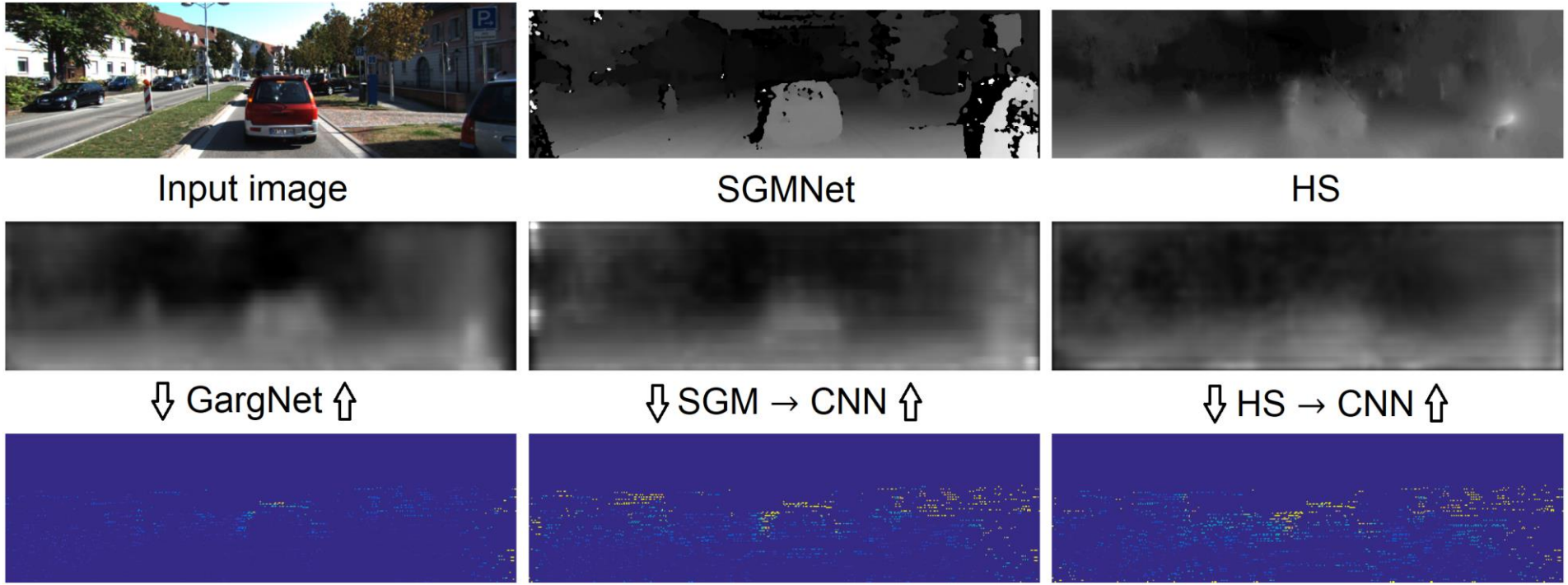
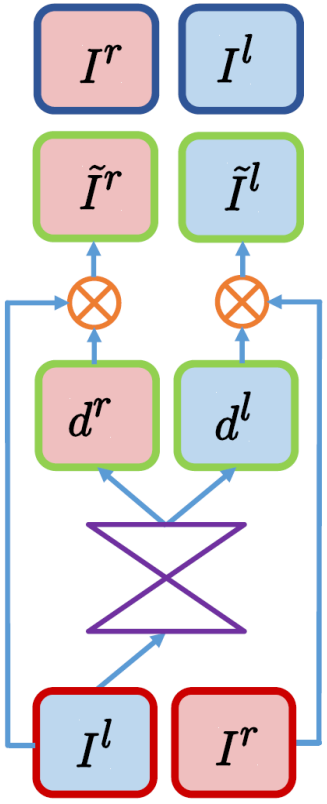


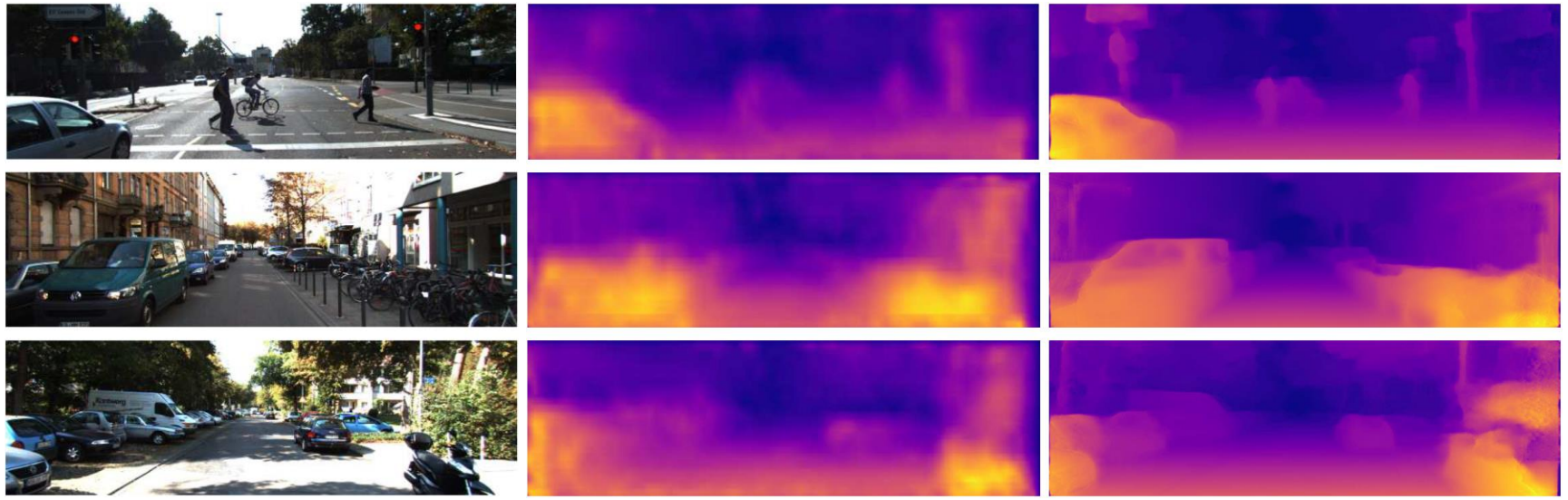
Figure slightly modified and taken from [49] showing depth results and errors (bottom row) of GargNet [49], SGMNet [44], HS algorithm [58] and SGM/HS derivatives on KITTI [48]

- **Monodepth [50]** – also view it as an image reconstruction problem
- Train the network on left image to produce both disparities to combat ‚texture-copy' artifacts
- Combine appearance matching loss using SSIM [59] L_{ap}, L1 disparity smoothness loss on disparity gradients L_{ds} and novel left-right consistency loss $L_{lr} = \frac{1}{N} \sum_{i,j} | d_{ij}^{l} - d_{ij+d_{ij}^{l}}^{r} |$ (22)

$$L_s = \alpha_{ap}\left(L_{ap}^{l} + L_{ap}^{r}\right) + \alpha_{ds}(L_{ds}^{l} + L_{ds}^{l}) + \alpha_{lr}(L_{lr}^{l} + L_{lr}^{l}) \quad (23)$$

See paper [50] for α – values

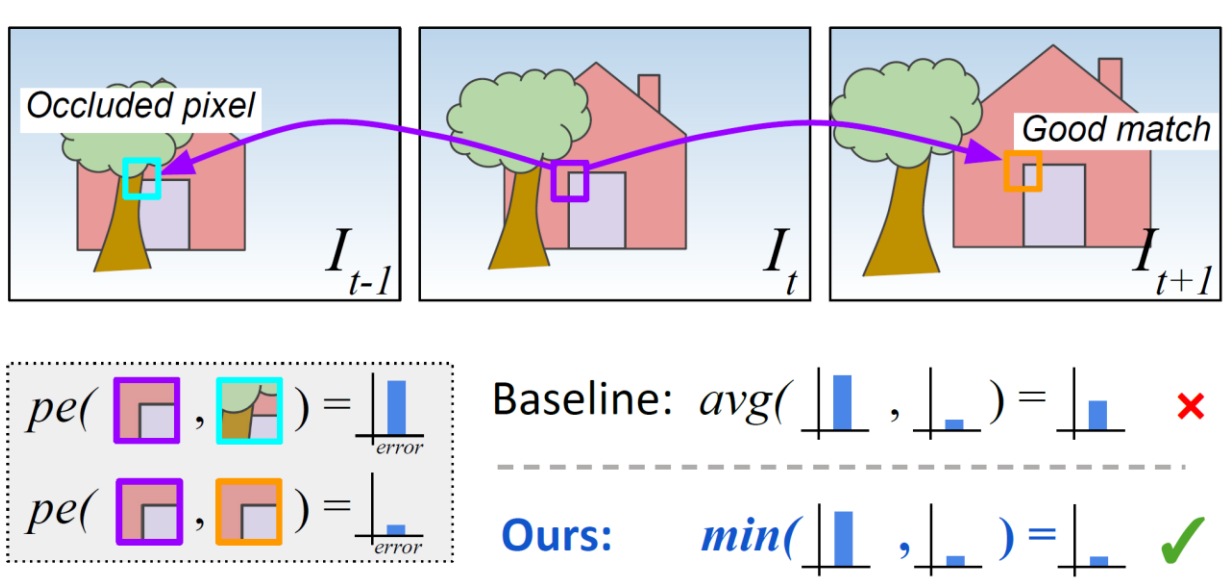




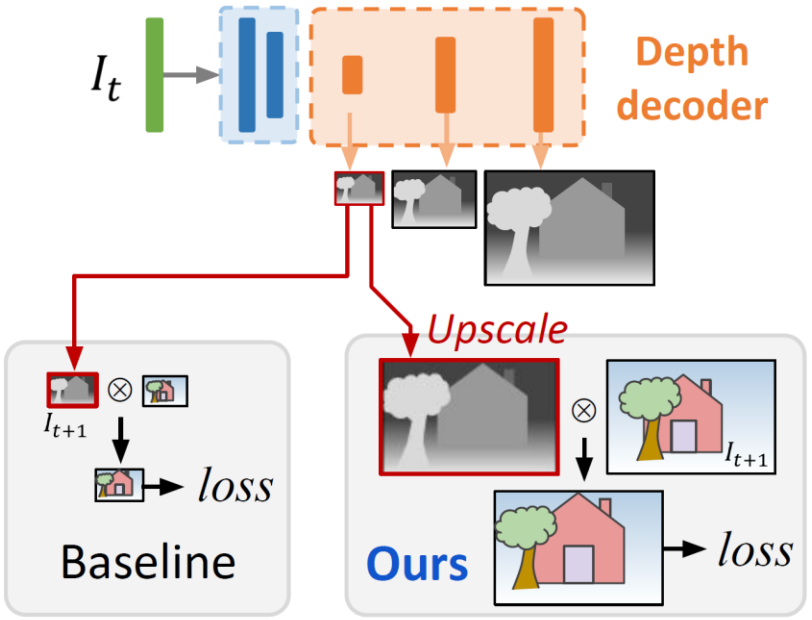
Left figure taken from and sampling strategy of Monodepth [50]

Figure modified and taken from [50] showing (left to right): KITTI Eigen [48] input images, GargNet [49] and Monodepth [50] quantitative results

- **Monodepth2 [56]** – self-supervision from stereo or sequence of images
- Use a standard U-Net [60] to predict depth
- Argue that avg. error matches occluded pixels while minimum does not
- Upscale depth prediction at multiple layers to compute cost at input resolution
- Mask stationary (relative to camera) pixels using temporal sequence



Appearance loss as minimum of projection errors



Upsample at multiple scales to input size

Figure modified and taken from [56] showing an overview of the method minimum reprojection and multi-scale loss calculation

- **Monodepth2 [56]** – sharper depth results using monocular (M) or stereo (S) supervision

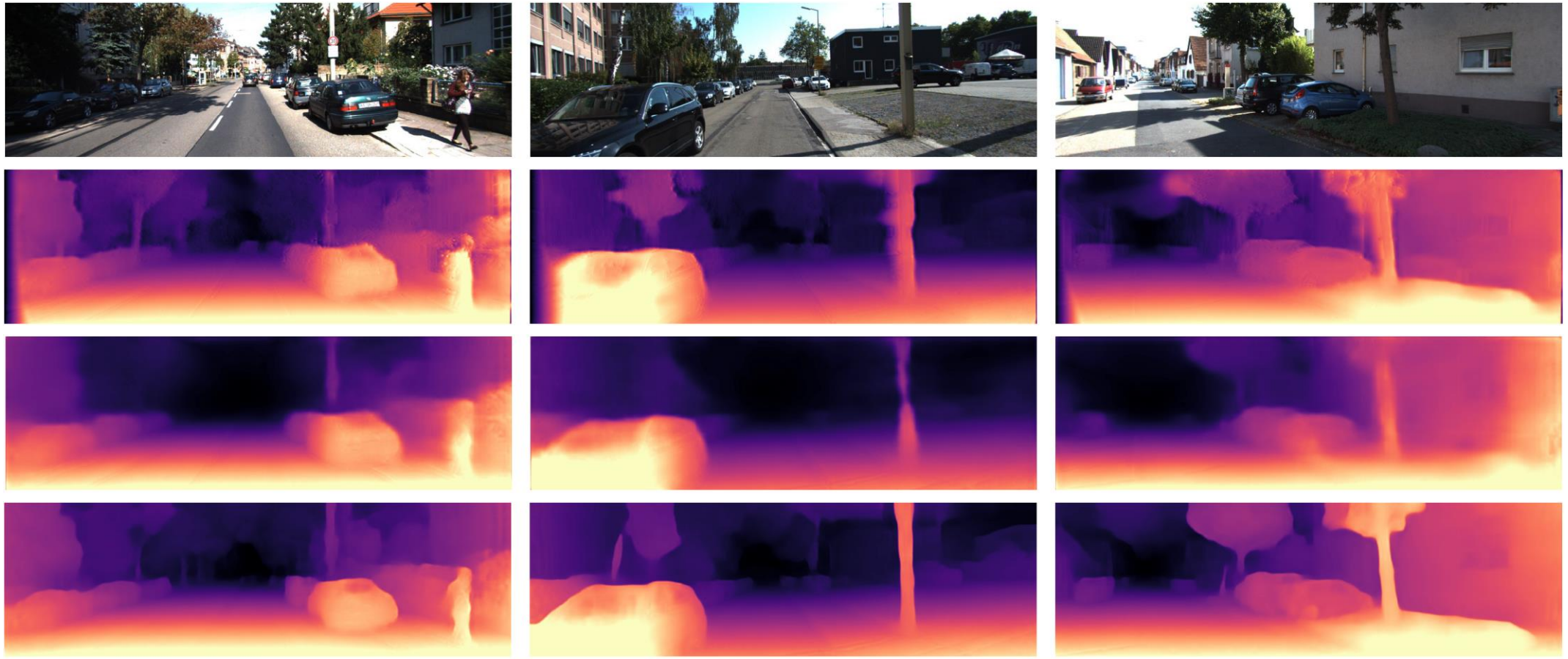


Figure modified and taken from [56] showing (top to bottom): KITTI Eigen [48] input images as well as quantitative results for Monodepth [50], EPC++(MS) [61] and Monodepth 2 [56] with monocular (M) supervision

- **HITNet [62]** – state-of-art results on ETH3D [63], Middlebury-v3 [64] and KITTI [48]

1. Uses a very small U-Net [60] for multi-resolution feature extraction – **supervised 2D**
2. Initialization stage to extract initial disparity d^{init} and feature vector $\boldsymbol{p}^{init}$
3. Slanted plane hypothesis with disparity d, gradients d_x, d_y and descriptor $\boldsymbol{h} = [d, d_x, d_y, \boldsymbol{p}]$
4. Propagation stage to refine disparity hypotheses with slanted support windows

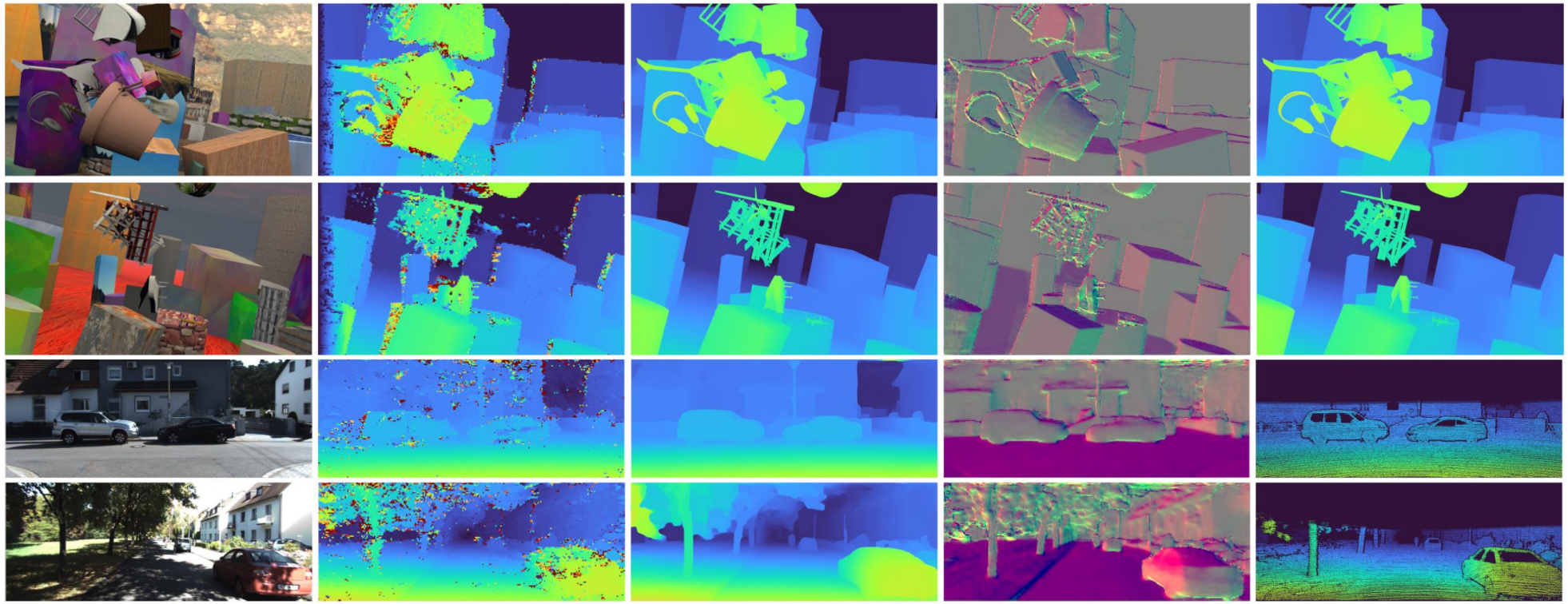


Figure taken from & results of HITNet [62] on SceneFlow [36] (top 2 rows) and KITTI [48] (bottom 2)

Left-to-right: input image, initialization, disparity result, predicted slant and ground truth

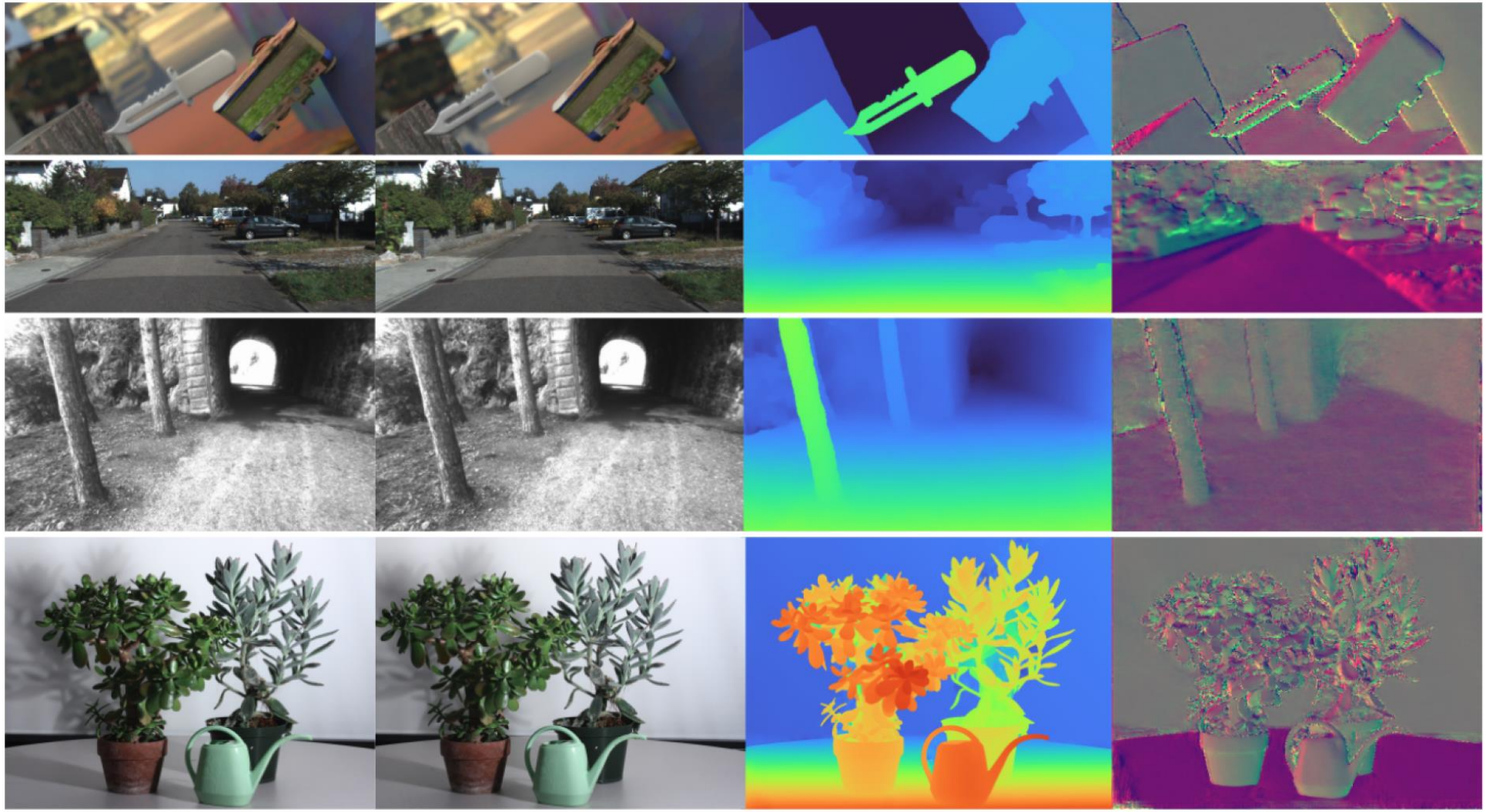


Figure taken from [62] showing HITNet results (disparity and slant) on (top-to-bottom) SceneFlow [36], KITTI [48], ETH3D [63] and Middlebury-v3 [64] - Using a Titan V GPU it runs 19ms per frame for ETH3D and KITTI (0.5Mpixel) and 108ms per Mpix for Middlebury-v3

Summary of stereo vision (learning-based)

- Classical stereo matching [30]
- NN-based cost calculation [33]
- End-to-end 2D CNNs [36]
- End-to-end 3D CNNs [37]
- Unsupervised monodepth [49]

- HITNet [62] already looks really close to ground-truth - is (DL-based) stereo vision solved?

- It still struggles with texture-less regions, requires dense LiDAR-data & expensive GPU...

- Outlook: what else is happening in machine-learning based computer vision?

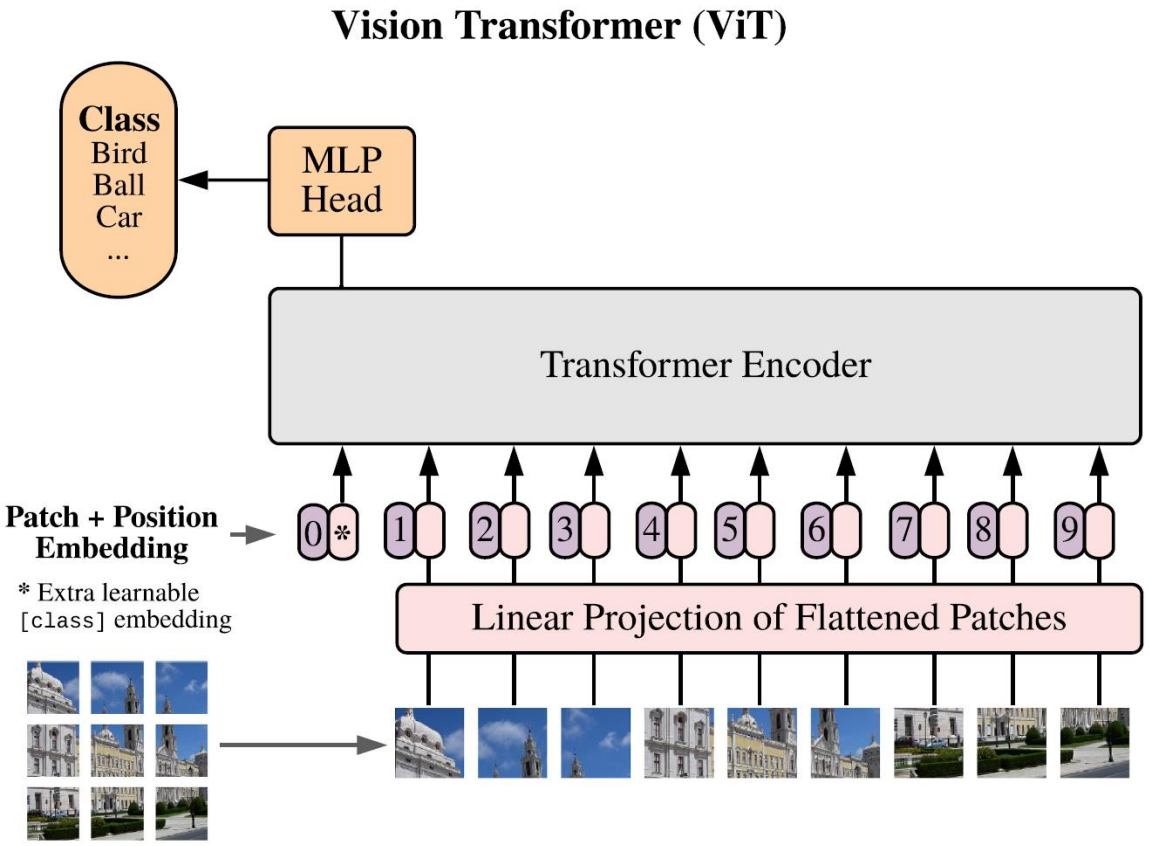
Outlook - Transformers

- Convolutions in CNNs go back to the age-old bias-variance dilemma [65]:
  - Size of convolutional kernel determines receptive field
  - Small (e.g. 3x3) capture local-context, biased
  - Largest reasonable is 7x7 – context can span much larger distances
  - Lack of global understanding, i.e. dependencies between features
  - How to capture long-range relations without increasing complexity?

- **Transformers (230+ preprints in last 5 months: git-repo)**
  - Taken from natural language processing (NLP) – embeds words in codebook
  - Two main ideas [66]:
    - **Self-attention** to capture long-range relations between embeddings
    - **Pre-training** on large (un)labelled corpus in (un)supervised manner
      → fine-tuning for target task on small labeled dataset

Transformers in computer vision

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale [67]:



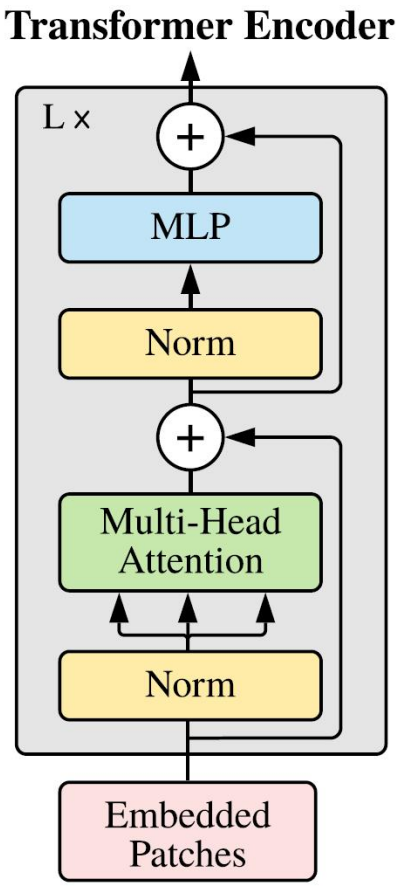
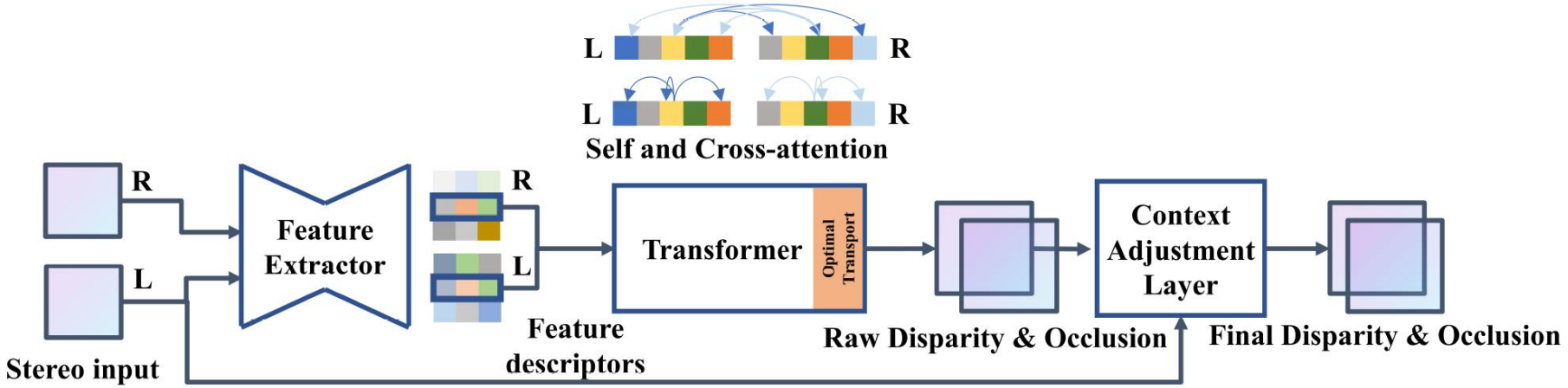


Image taken from [67] where the illustration of the transformer encoder was inspired by [68]

1. Image is split into patches
2. Embedded with position
3. Fed into transformer
4. Attention & dependencies
5. Classification with learnable “classification token”

Stereo vision with transformers: STTR [69]



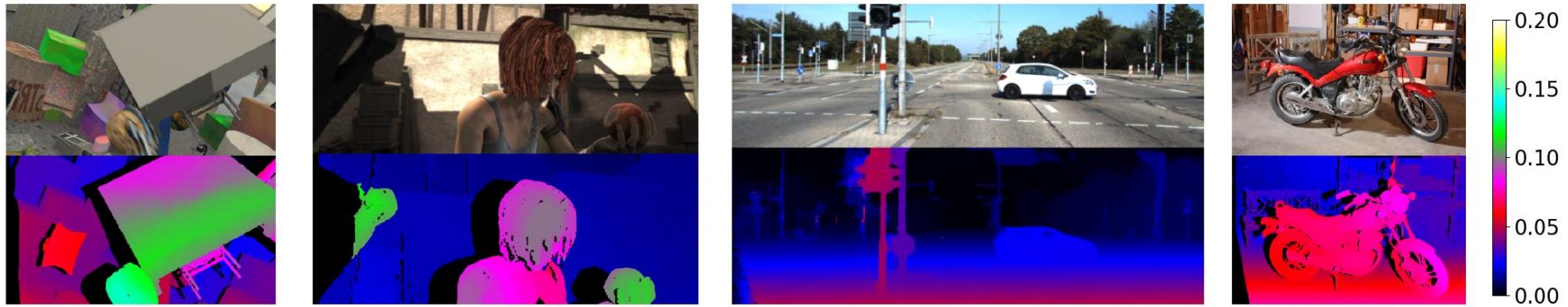


Figure taken from STTR [67] showing the network overview (top), left input images (middle) and inference results (bottom) of a model trained only on SceneFlow [36] - datasets from left to right: SceneFlow [36], MPI Sintel [70], KITTI [48] and Middlebury-v3 [64] - color map is relative to image width with the scale given at the right side

Mono-depth transformers: MiDaS [71] & DPT [72]

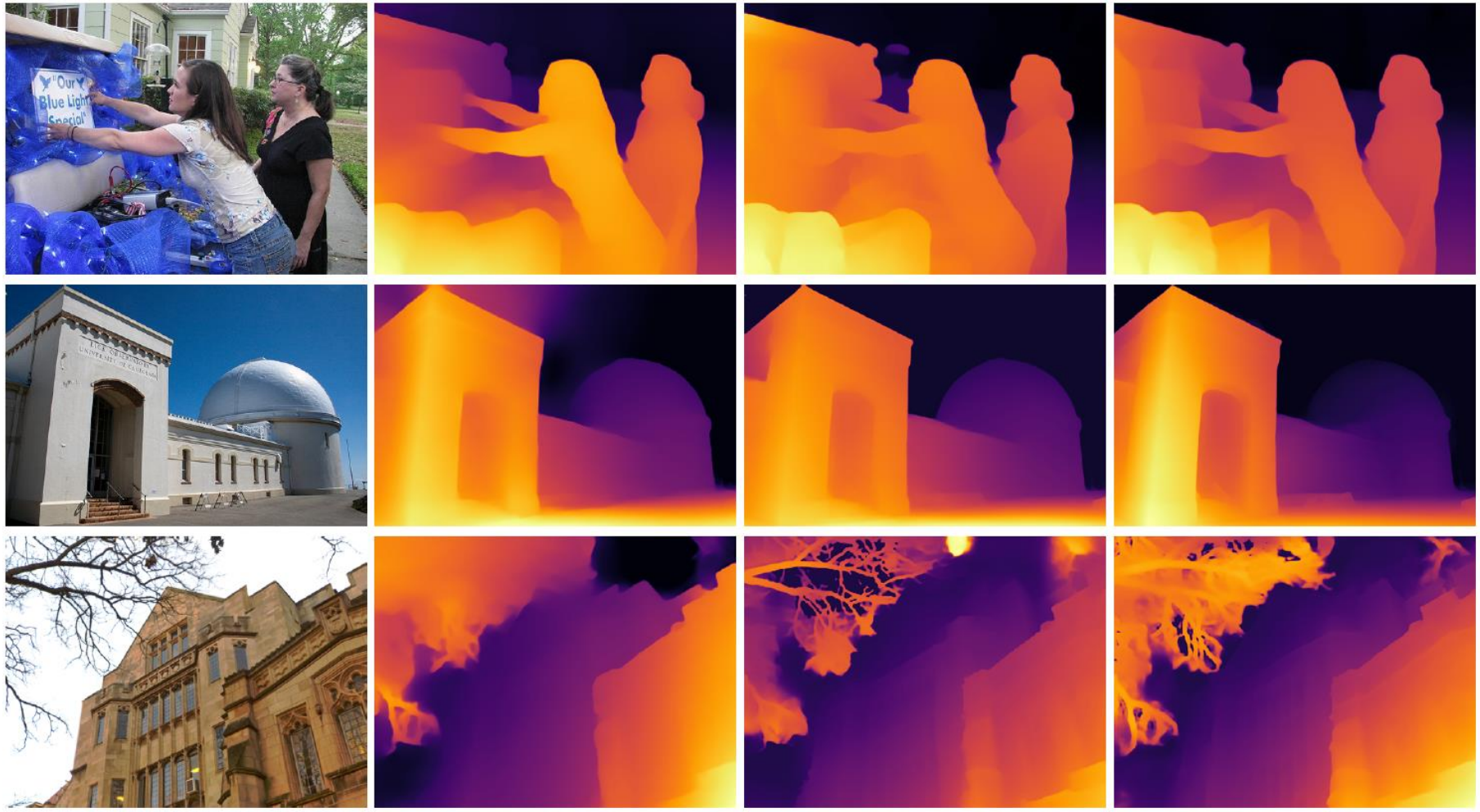


Figure taken from DPT [72]

Left-to-right: input images (source unknown), MiDaS [71], DPT-Hybrid and DPT-Large [72] results

These are zero-shot cross-dataset transfers, i.e. inferences in **datasets** the model has not seen

Models were trained on *MIX 6*, a meta-dataset composed of 1.4 million images for monocular depth estimation [72]

Questions?

Suggestions?

Complaints?

Thank you for your attention

References

References marked with * have not been read by the lecturer (A. Kriegler) but are instead included for the sake of providing a historically accurate representation of the scientific progress in the field of deep learning. The correctness of this representation is largely based on works by J. Schmidhuber [2, 75].

[1] C.M. Bishop, *Pattern Recognition and Machine Learning,* Singapore: Springer, 2006. (PDF)

[2] J. Schmidhuber, “Deep learning in neural networks: An overview”, *Neural networks,* 2015, 61, pp.85-117, 2015. (PDF)

[3] I. Goodfellow, Y. Bengio and A. Courville, *Deep learning*, Cambridge: MIT press, 2016. (e-Book)

[4] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach,* Fourth edition, New Jersey: Pearson, 2021. (accompanying website)

[5] S. Thrune, W. Burgard and D. Fox, *Probabilistic Robotics,* Cambridge: MIT Press, 2005. (accompanying website)

[6] T. Gärtner, M. Thiessen and A. Sepliarskaia, *194.100 Theoretical Foundations and Research Topics in Machine Learning.* [Online]. Available at: https://preview.tinyurl.com/yf73xbu7 (Accessed: 11.05.2021).

[7] M. Charikar and C. Ré, *CS229: Machine Learning.* [Online]. Available at: http://cs229.stanford.edu (Accessed: 11.05.2021).

[8] L. FeiFei, K. Ranjay, X. Danfei, *CS231n: Convolutional Neural Networks for Visual Recognition.* [Online]. Available at: http://cs231n.stanford.edu (Accessed: 11.05.2021).

[9] L. Fridman, *MIT Deep Learning and Artificial Intelligence Lectures.* [Online]. Available at: https://deeplearning.mit.edu (Accessed: 11.05.2021).

[10] G. Sanderson, *Deep Learning Series.* [Online]. Available at: https://preview.tinyurl.com/m92b8r9u (1st part) (Accessed: 11.05.2021).

[11] U. Von Luxburg, *Mathematics for Machine Learning.* [Online]. Available at: https://preview.tinyurl.com/jk9p3m9x (Accessed: 11.05.2021).

*[12] A. Ivakhnenko and V.G. Lapa, *Cybernetic predicting devices,* New York: CCM Information Corp., 1965. (accompanying website)

*[13] J. Schmidhuber, “Learning Complex, Extended Sequences using the Principle of History Compression”, *Neural Computation*, 1992, 4(2), pp.234-242, 1992. (PDF)

[14] S. Hochreiter, *Untersuchungen zu Dynamischen Neuronalen Netzen*. München: Technische Universität München (master‘s thesis, german), 1991. (PDF)

[15] A. Krizhevsky, I. Sutskever and G.E. Hinton, “Imagenet Classification with Deep Convolutional Neural Networks”, *Advances in Neural Information Processing Systems (NIPS)*, 2015, 25, pp.1097-1105, 2015. (PDF)

References

[16] W. Fedus, B. Zoph and N. Shazeer, *Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity*. [Online]. Available at: https://arxiv.org/abs/2101.03961 (Accessed: 11.05.2021).

[17] Ž. Ivezić, A.J. Connolly, J.T. Vanderplas and A. Gray, *Statistics, Data Mining and Machine Learning in Astronomy,* New Jersey: Princeton University Press, 2014. (accompanying website)

[18] S. Linnainmaa, The Representation of the Cumulative Rounding Error of an Algorithm as a Taylor Expansion of the Local Rounding Errors.* Helsinki: University of Helsinki (master's thesis, finnish)*,* 1970. (following Springer paper)

*[19] L. Euler, *Methodus Inveniendi Lineas Curvas Maximi Minimive Proprietate Gaudentes,* Lausanne & Geneva: Marcum-Michaelem Bousquet, 1744. (euler archive)

*[20] K. Fukushima and S. Miyake, "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition", *Biological Cybernetics,* 1980, 36, pp. 193-202, 1980. (PDF)

[21] S. Saha, *A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way.* [Online]. Available at: https://preview.tinyurl.com/6d8njrv6 (Accessed: 11.05.2021).

[22] Aphex34, *typical CNN architecture.* [Online]. Available at: https://preview.tinyurl.com/33xvmpfm (Accessed: 11.05.2021).

[23] M.D. Zeiler and R. Fergus, „Visualizing and Understanding Convolutional Networks", *European Conference on Computer Vision*, 2014, pp.818-833, 2014. (arXiv preprint)

[24] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, „Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp.618-626, 2017. (PDF)

[25] J.T. Springenberg, A. Dosovitskiy, T. Brox and M. Riedmiller, „Striving for Simplicity: The All Convolutional Net", ICLR (workshop track), 2015. (accompanying website)

[26] J. Duchi, „Introduction to Convex Optimization for Machine Learning", *Practical Machine Learning, Fall 2009*, UC Berkeley, 2009. (PDF)

[27] H. Dawar, *Stochastic Gradient Descent.* [Online]. Available at: https://preview.tinyurl.com/erz8k26h (Accessed: 11.05.2021).

[28] L. Von Rueden, S. Mayer, R. Sifa, C. Bauckhage and J. Garcke, „Combining Machine Learning and Simulation to a Hybrid Modelling Approach: Current and Future Directions", *International Symposium on Intelligent Data Analysis (IDA)*, 2020, pp.548-560, 2020. (PDF)

References

[29] Original source unknown. Taken from: L. Langer, *Algorithm Journey from PC to IoT – A Signal Processing Pipeline with TensorFlow 2.X on a Smartwatch.* [Online]. Available at: https://preview.tinyurl.com/5xfkchpb (Accessed: 11.05.2021).

[30] D. Scharstein and R. Szeliski, „A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms“, *International Journal of Computer Vision*, 2002, 47 (1/2/3), pp.7-42, 2002. (PDF)

[31] M.S. Hamid, N.A. Manap, R.A. Hamzah and A.F. Kadmin, „Stereo matching algorithm based on deep learning: A survey“, *Journal of King Saud University – Computer and Information Sciences*, 32, 2020. (ScienceDirect)

[32] K. Zhou, X. Meng and B. Cheng, „Review of Stereo Matching Algorithms Based on Deep Learning“, *Computational Intelligence and Neuroscience*, Volume 2020, 2020. (PDF)

[33] J. Zbontar and Y. LeCun, „Computing the Stereo Matching Cost with a Convolutional Neural Network“, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2015, pp.1592-1599, 2015. (PDF)

[34] Z.Y. Chen, X. Sun, L. Wang, Y.A. Yu and C. Huang, „A Deep Visual Correspondence Embedding Model for Stereo Matching Costs“, *Proceedings of the International Conference on Computer Vision (ICCV),* 2015, pp.972-980, 2015. (PDF)

[35] W. Luo, A. Schwing and R. Urtasun, „Efficient Deep Learning for Stereo Matching“, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp.5695-5703, 2016. (PDF)

[36] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy and T. Brox, „A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation“, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2016, pp.4040-4048, 2016. (PDF)

[37] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy and A. Bachrach, „End-to-End Learning of Geometry and Context for Deep Stereo Regression“, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp.66-75, 2017. (PDF)

[38] J. Flynn, I. Neulander, J. Philbin and N. Snavely, „DeepStereo: Learning to Predict New Views from the World‘s Imagery“, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recoginition (CVPR),* 2016, pp.5515-5524, 2016. (PDF)

[39] J.Y. Xie, R. Girshick and A. Farhadi, „Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks“, *Proceedings of the European Conference on Computer Vision (ECCV),* 2016, pp.842-857, 2016. (PDF)

References

[40] A.T. Eitan, E. Smolyansky, I.K. Harpaz and S. Perets, *Connected Papers.* [Online]. Available at: https://www.connectedpapers.com (Accessed: 11.05.2021).

[41] X. Han, T. Leung, Y. Jia, R. Sukthankar and A.C. Berg, „MatchNet: Unifying Feature and Metric Learning for Patch-Based Matching“, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2015, pp.3279-3286, 2015. (PDF)

[42] S. Zagoruyko and N. Komodakis, „Learning to Compare Image Patches via Convolutional Neural Networks“, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2015, pp. 4353-4361, 2015. (PDF)

[43] J. Zbontar and Y. LeCun, „Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches“, *Journal of Machine Learning Research,* 17 (2016), pp.1-32, 2016. (PDF)

[44] A. Seki and M. Pollefeys, „SGM-Nets: Semi-global matching with neural networks“, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp.231-240, 2017. (PDF)

[45] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers and T. Brox, „FlowNet: Learning Optical Flow with Convolutional Networks, *Proceedings of the IEEE International Conference on Computer Vision (ICCV),* 2015, pp.2758-2766, 2015. (PDF)

[46] Z. Liang, Y. Feng, Y. Guo, H. Liu, W. Chen, L. Qiao, L. Zhou and J. Zhang, „Learning for Disparity Estimation through Feature Constancy“, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2811-2820, 2018. (PDF)

[47] G. Yang, H. Zhao, J. Shi, Z. Deng and J. Jia, „SegStereo: Exploiting Semantic Information for Disparity Estimation“, *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 636-651, 2018. (PDF)

[48] M. Menze, C. Heipke and A. Geiger, „Object Scene Flow.“ *ISPRS Journal of Photogrammetry and Remote Sensing*, 2018, 140, pp.60-76. (PDF)

[49] R. Garg, V. Kumar, and I. Reid, „Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue“, *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 740-756, 2016. (PDF)

[50] C. Godard, O. MacAodha and G.J. Brostow, „Unsupervised Monocular Depth Estimation with Left-Right Consistency“, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp.270-279, 2017. (PDF)

[51] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, „The Cityscapes Dataset for Semantic Urban Scene Understanding“, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp.3213-3223, 2016. (PDF)

References

[52] S. Khamis, S. Fanello, C. Rhemann, A.Kowdle, J. Valentin and S. Izadi, „StereoNet: Guided Hierarchical Refinement for Real-Time Edge-Aware Depth Prediction“, *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp.573-590. (PDF)

[53] J.R. Chang and Y.S. Chen, „Pyramid Stereo Matching Network“, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp.5410-5418, 2018. (PDF)

[54] A.G. Kendall, *Geometry and Uncertainty in Deep Learning for Computer Vision*, Cambridge: University of Cambridge (doctoral dissertation), 2018. (PDF)

[55] D. Eigen, C. Puhrisch and R. Fergus, „Depth Map Prediction from a Single Image using a Multi-Scale Deep Network“, *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp.2366-2374, 2014. (PDF)

[56] C. Godard, O.M. Aodha, M. Firman and G.J. Brostow, „Digging Into Self-Supervised Monocular Depth Estimation“, *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp.3828-3838, 2019. (PDF)

[57] H. Hirschmüller, „Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information“, *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),* 2005, Vol.2, pp.807-814, 2005. (PDF)

[58] B.K. Horn and B.G. Schunck, „Determining Optical Flow“, *Artificial Intelligence*, 17, 1-3, pp.185-203, 1981. (PDF)

[59] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli, „Image Quality Assessment: From Error Visibility to Structural Similarity“, *Transactions on Image Processing*, 13(4), pp.600-612, 2004. (PDF)

[60] O. Ronneberger, P. Fischer and T. Brox, „U-Net: Convolutional Networks for Biomedical Image Segmentation“, *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015, Springer, pp.234-241, 2015. (PDF)

[61] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia and Y. Yuille, „Every Pixel Counts++: Joint Learning of Geometry and Motion with 3d Holistic Understanding“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10), pp.2624-2641, 2019. (PDF)

[62] V. Tankovich, C. Häne, Y. Zhang, A. Kowdle, S. Fanello and S. Bouaziz, *HITNet: Hierarchical Iterative Tile Refinement Network for Real-time Stereo Matching*. [Online] Available at: https://arxiv.org/abs/2007.12140 (Accessed: 11.05.2021).

[63] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, A. Geiger, "A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2017, pp.3260-3269, 2017. (PDF)

References

[64] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang and P. Westling, „High-Resolution Stereo Datasets with Subpixel-Accuracte Ground Truth“, *German Conference on Pattern Recognition (GCPR)*, 2014, pp.31-42, 2014. (PDF)

[65] S. Geman, E. Bienenstock and Rene Doursat, „Neural Networks and the Bias/Variance Dilemma“, *Neural Computation*, 4(1), 1-58, 1992. (PDF)

[66] S. Khan, M. Naseer, M. Hayat, S.W. Zamir, F.S. Khan and M. Shah, *Transformers in Vision: A Survey*. [Online]. Available at: https://arxiv.org/abs/2101.01169 (Accessed: 11.05.2021).

[67] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly and J. Uszkoreit, *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. [Online]. Available at: https://arxiv.org/abs/2010.11929 (Accessed: 11.05.2021).

[68] A. Vaswani, N. Shazeer, N. Parmar, J. Uskoreit, L. Jones, A.N. Gomez, L. Kaiser and I. Polosukhin, „Attention Is All you Need“, *Proceeding of the 31st International Conference on Neural Information Processing (NIPS)*, 2017, pp.6000-6010, 2017. (PDF)

[69] Z. Li, X. Liu, N. Drenkow, A. Ding, F.X. Creighton, R.H. Taylor, M. Unberath, *Revisiting Stereo Depth Estimation From a Sequence-to-Sequence Perspective with Transformers*. [Online]. Available at: https://arxiv.org/abs/2011.02910 (Accessed: 11.05.2021).

[70] D.J. Butler, J. Wulff, G.B. Stanley and M.J. Black, „A Naturalistic Open Source Movie for Optical Flow Evaluation“, *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012, pp.611-625, 2012. (PDF)

[71] R. Ranftl, K. Lasing, D. Hafner, K. Schindler and V. Koltun, „Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer“, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, doi: 10.1109/TPAMI.2020.3019967, 2020. (arXiv)

[72] R. Ranftl, A. Bochkovskiy and V. Koltun, *Vision Transformers for Dense Prediction*. [Online]. Available at: https://arxiv.org/abs/2103.13413 (Accessed: 11.05.2021).

*[74] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Francisco California: Morgan Kaufmann Publishers, 1988. (PDF)

[75] J. Schmidhuber, *Critique of 2018 Turing Award for Drs. Bengio & Hinton & LeCun.* [Online]. Available at: https://preview.tinyurl.com/3vmfemxh (Accessed: 11.05.2021).