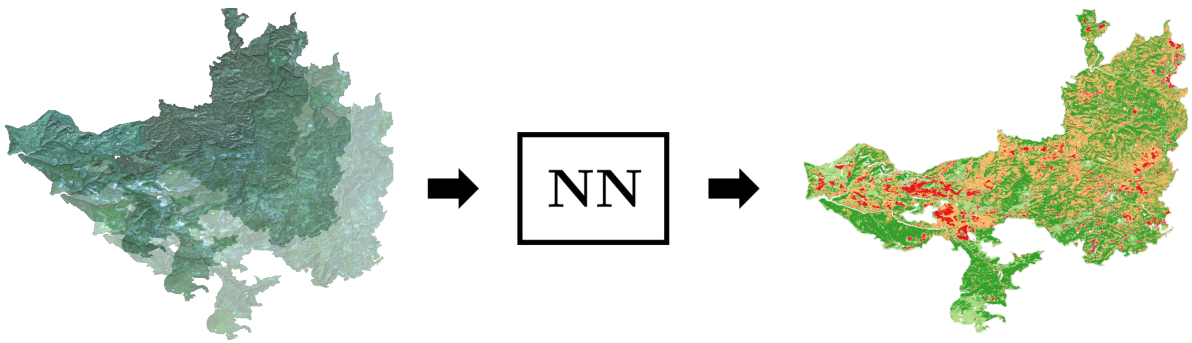
Semi-Supervised Learning for Spatio-Temporal Segmentation of Satellite Images

Antonín Hruška

June 5, 2023

Motivation

ESA project[1] in collaboration with Mapradix s.r.o.
Enhanced Spatiotemporal Land Change Monitoring Based on Sentinel-2 Time Series and VHR Images



[1] At https://eo4society.esa.int/projects/spatiotemporal-sen2vhr/

Remote Sensing and its issues

Sentinel-2 imagery

- 13 spectral bands
- period of 10 days
- different spatial resolution ($10/20/60\ m^2$)

Issues

- Partial Oclusions: Cloud cover or snow cover (seasonal dependence)
- Different viewing angle
- Atmospheric calibration etc.
- Annotation process is time demanging and requires high expertise.

Semi Supervised Learning can reduce the need for large number of annotations. Generative models can impaint the ocluded regions.

Semi Supervised Learning

$$\mathcal{D} = \mathcal{X} \cup \mathcal{U} : \mathcal{X} = \{(x_1, y_1), \ldots, (x_l, y_l)\}, \quad \mathcal{U} = \{(x_{l+1}), \ldots, (x_u)\},$$

where x_i are features and y_i are labels.
Assumptions from unsupervised learning:

- Smothness: x_1 close to x_2 in high density region imples y_1 close to y_2
- Cluster: Points in one cluster are likely to be of the same class
- Manifold: The data lie along low-dimensional latent manifolds inside that high-dimensional space.

MixMatch

Discriminative model based on empirical risk minimization

- Data Augmentation
- Pseudo labeling
- Sharpening
- MixUp

$$\mathcal{X}', \mathcal{U}' = \text{MixMatch}(\mathcal{X}, \mathcal{U}, T, K, \alpha)$$

$$\mathcal{L}_{\mathcal{X}} = \frac{1}{|\mathcal{X}'|} \sum_{x', p' \in \mathcal{X}'} H(p', f_\theta(x')), \quad \mathcal{L}_{\mathcal{U}} = \frac{1}{L|\mathcal{U}'|} \sum_{u', q' \in \mathcal{U}'} ||q' - f_\theta(u')||_2^2$$

$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_{\mathcal{U}} \mathcal{L}_{\mathcal{U}}$$

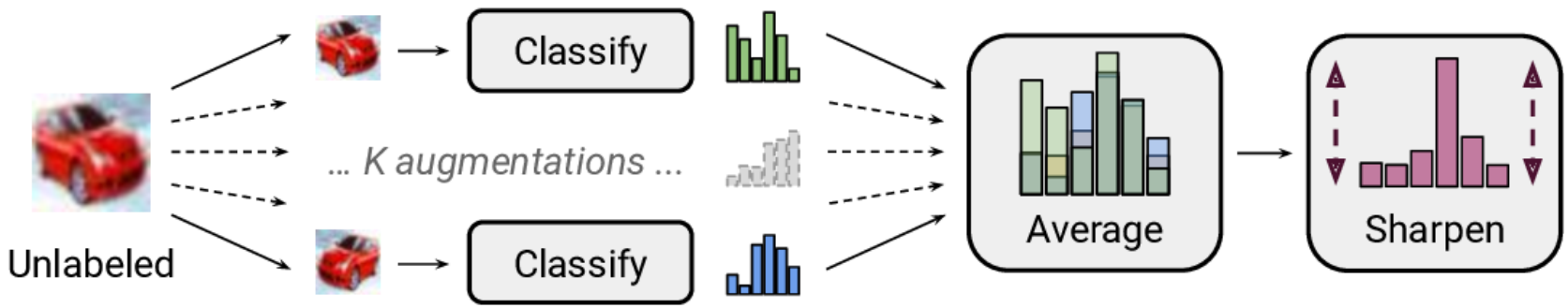


Figure: Mixmatch up to the point of sharpening. Source: MixMatch

MixMatch: A Holistic Approach to Semi-Supervised Learning

MixMatch

After sharpening :

$$\mathcal{X}^\star = \{(\tilde{x}_i, p_i) \,|\, i \in \{1, \ldots, n\}\},\ |\mathcal{X}^\star| = n$$
$$\mathcal{U}^\star = \{(\tilde{u}_{j,k}, q_j) \,|\, j \in \{1, \ldots, n\},\ k \in \{1, \ldots, K\}\},\ |\mathcal{U}^\star| = Kn$$
$$\mathcal{W} = \text{Shuffle}(\text{Concat}(\mathcal{X}^\star, \mathcal{U}^\star))$$

MixUp:

$$\begin{bmatrix} \mathcal{X}' \\ \mathcal{U}' \end{bmatrix} = \text{MixUp}(\begin{bmatrix} \mathcal{X}^\star \\ \mathcal{U}^\star \end{bmatrix}, \mathcal{W})$$

where the MixUp$(\cdot,\cdot)$ corresponds to following operation applied elementwise:

$$\lambda' = \max(\lambda, 1-\lambda),\ \text{where } \lambda \sim \text{Beta}(\alpha, \alpha)$$
$$\begin{bmatrix} x' \\ p' \end{bmatrix} = \lambda' \begin{bmatrix} x_1 \\ p_1 \end{bmatrix} + (1-\lambda') \begin{bmatrix} x_2 \\ p_2 \end{bmatrix}$$

VAE

Generative model based on ML maximization
Simple bayesian network with hidden variables z:

$$p_\theta(x, z) = p_\theta(z)p_\theta(x \mid z)$$

where $p_\theta(z)$ or $p_\theta(x \mid z)$ is specified and parametrized by NN.
The intractability of evidence:

$$p_\theta(x) = \int_z p_\theta(x, z)\,\mathrm{d}z = \frac{p_\theta(x, z)}{p_\theta(z|x)}$$

do not allow us to maximize the log-likelihood directly, instead we introduce the *inference model* $q_\phi(z|x)$ and optimize the ELBO proxy:

$$\begin{aligned}\log p_\theta(x) &= \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x, z)q_\phi(z|x)}{q_\phi(z|x)p_\theta(z|x)}\right] \\ &= \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x, z)}{q_\phi(z|x)}\right]}_{\mathrm{ELBO}} + \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[\log \frac{q_\phi(z|x)}{p_\theta(z|x)}\right]}_{\mathrm{KL}(q_\phi(z|x)||p_\theta(z|x))}\end{aligned}$$

VAE

Learning : Stochastic approach - unbaiessed estimator and reparametrization trick allow us to use SGD

$$\mathcal{L} = \mathbb{E}_{q_\phi(z|x)}\Big[\log p_\theta(x,z)\Big] - \mathbb{E}_{q_\phi(z|x)}\Big[\log q_\phi(z|x)\Big]$$

Issues and limitations:

- Parametrization trick requires continouous latent variable
- Posterior collapse: Optimization can end in an undesirable stable equilibrium: $q_\phi(z|x) \approx p_\theta(z)$
- Bluriness of images: The ELBO enforces the consistent encoder and decoder pairs. This disallow the model to learn the complex data distribution. Solved by more flexible model (such as hiearchical VAE).

Symmetric equillibrium leaning in VAE

New objectives are

$$\begin{aligned}\mathcal{L}_p(\theta,\phi) &= \mathbb{E}_{\pi(x,z)}\big[\log p_\theta(x,z)\big] + \mathbb{E}_{\pi(z)}\big[\log p_\theta(z)\big] + \\ &\quad \mathbb{E}_{\pi(x)}\,\mathbb{E}_{q_\phi(z|x)}\big[\log p_\theta(x,z)\big] \\ \mathcal{L}_q(\theta,\phi) &= \mathbb{E}_{\pi(x,z)}\big[\log q_\phi(z|x)\big] + \mathbb{E}_{\pi(z)}\,\mathbb{E}_{p_\theta(x|z)}\big[\log q_\phi(z|x)\big]\end{aligned}$$

for semi-supervised training instead of ELBO:

$$\mathbb{E}_{\pi(x)}\big[\log p_\theta(x) - \mathrm{KL}(q_\phi(z|x)||p_\theta(z|x))\big]$$

The optimization does not require the reparametrization trick and allows for broader families of distributions, such as exponential family. *Symmetric Equilibrium Learning of VAE* [submitted to NeurIPS 23]

Methods

- CityScape dataset (simple 2D segmentation)
- U-net backbone
- plain accuracy and IoU metric
- hiearchical VAE for symmetric equilibrium learning

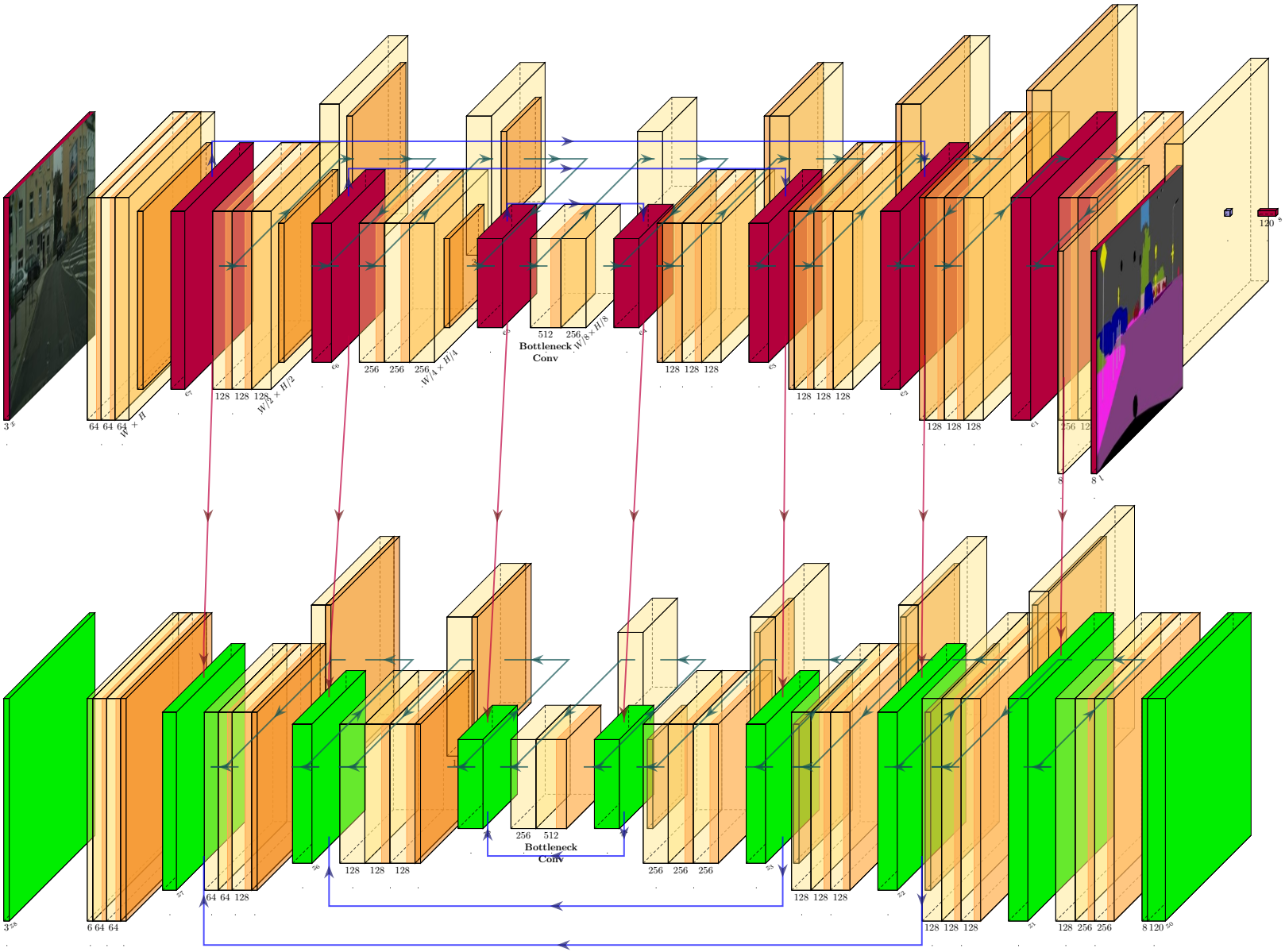
$$\mathsf{z} = (\mathsf{z}_0, \mathsf{z}_1, \ldots, \mathsf{z}_m), \quad \mathsf{z}_0 = (\mathsf{s}, \mathsf{l}),\ \mathsf{z}_m = \mathsf{x}$$

with blockwise optimization:

$$\begin{aligned}
\mathcal{L}_p(\boldsymbol{\theta}, \boldsymbol{\phi}) &= \mathbb{E}_{\pi(\mathsf{x},\mathsf{s})}\, \mathbb{E}_{q_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathsf{z}_{>0},\mathsf{l}|\mathsf{x},\mathsf{s})} \left[\log p_{\boldsymbol{\theta}}(\mathsf{x}, \mathsf{z})\right] \\
&\quad + \mathbb{E}_{\pi(\mathsf{x})}\, \mathbb{E}_{q_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathsf{z}|\mathsf{x})} \left[\log p_{\boldsymbol{\theta}}(\mathsf{x}, \mathsf{z})\right] \\
\mathcal{L}_q(\boldsymbol{\theta}, \boldsymbol{\phi}) &= \mathbb{E}_{\pi(\mathsf{x},\mathsf{s})} \left[\log q_{\boldsymbol{\phi}}(\mathsf{s} \mid \mathsf{x})\right] \\
&\quad + \mathbb{E}_{\pi(\mathsf{s})}\, \mathbb{E}_{p(\mathsf{l})}\, \mathbb{E}_{p_{\boldsymbol{\theta}}(\mathsf{x},\mathsf{z}_{>0}|\mathsf{z}_0)} \left[\log q_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathsf{z}|\mathsf{x})\right]
\end{aligned}$$

$\pi(\mathsf{x}, \mathsf{s})$ represents the underlying distribution with marginals $\pi(\mathsf{x})$ and $\pi(\mathsf{s})$.

Architecture of the HVAE



Mixmatch results (tables)

Labels [#]	10	100	500	1000	All
Mixmatch	84.53	90.76	93.42	94.32	94.84
Supervised	76.85	87.59	93.50	94.71	95.58

Table: Mixmatch accuracy rate (%) on CityScape dataset

Labels [#]	10	100	500	1000	All
Mixmatch	48.10	61.98	69.24	71.97	73.08
Supervised	41.78	54.38	68.25	71.93	73.84

Table: Mixmatch average IoU (%) on CityScape dataset

Mixmatch results (images)

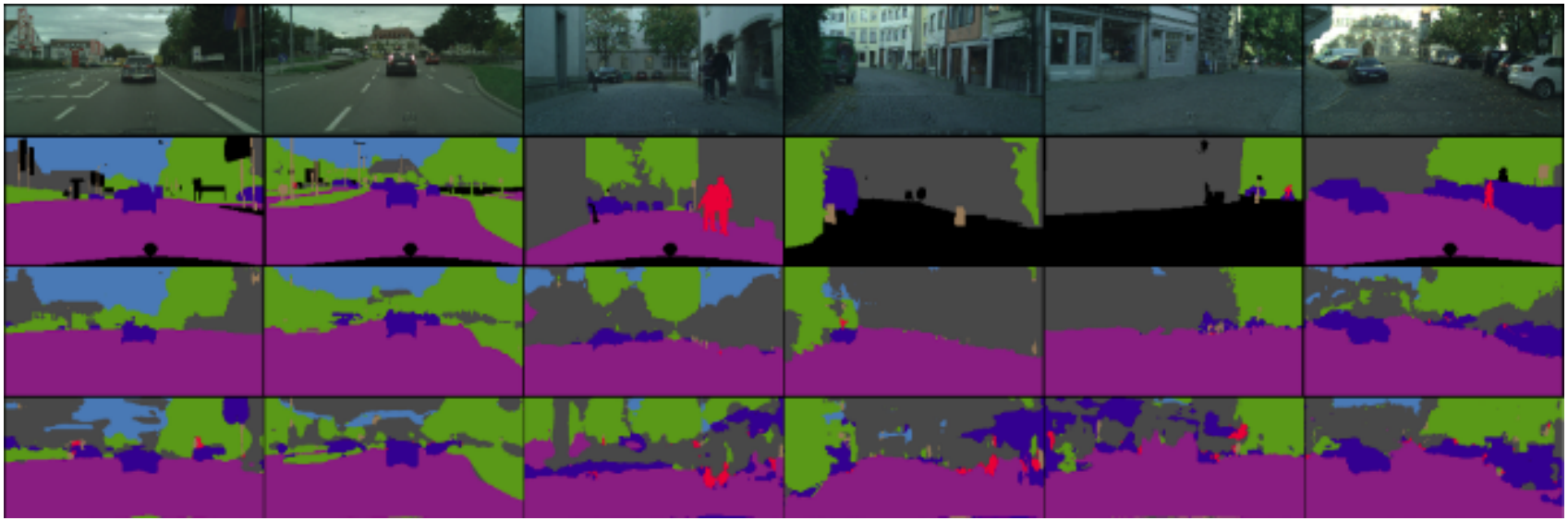


Figure: Models were trained on **ten** images. The first two rows contain the image and its ground truth segmentation. The MixMatch predictions are shown in the third row, while the supervised baseline predictions are in the fourth.
The colors used in the segmentation are as follows: flat (purple), human (red), vehicle (dark blue), construction (dark grey), object (light gray), nature (green), sky (light blue), and void (black). The predictions on the "void" class are not penalized nor evaluated.

Mixmatch results (images)

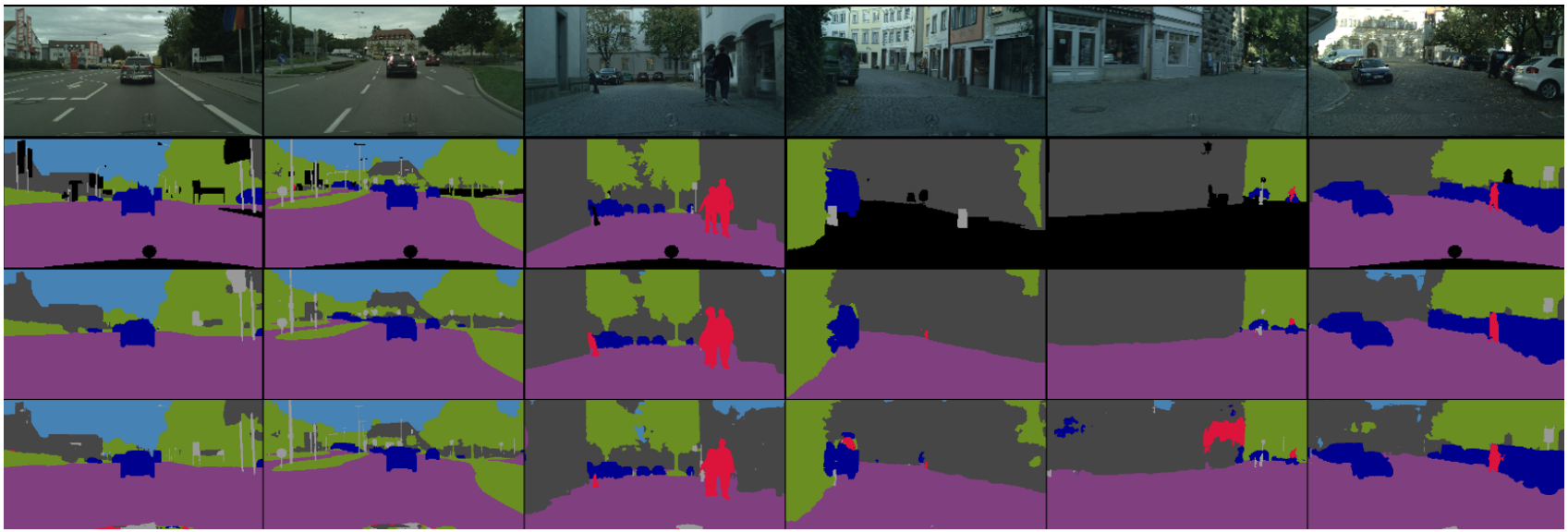


Figure: Models were trained on **all** images. The first two rows contain the image and its ground truth segmentation. The MixMatch predictions are shown in the third row, while the supervised baseline predictions are in the fourth.
The colors used in the segmentation are as follows: flat (purple), human (red), vehicle (dark blue), construction (dark grey), object (light gray), nature (green), sky (light blue), and void (black). The predictions on the "void" class are not penalized nor evaluated.

Symmetric equilibrium training results

Labels [#]	10	100	500	1000	All
HVAE	74.01	86.18	90.23	92.63	93.51
Supervised	74.82	87.73	90.81	44.05*	43.37*

Table: HVAE accuracy rate (%), * (learning has failed)

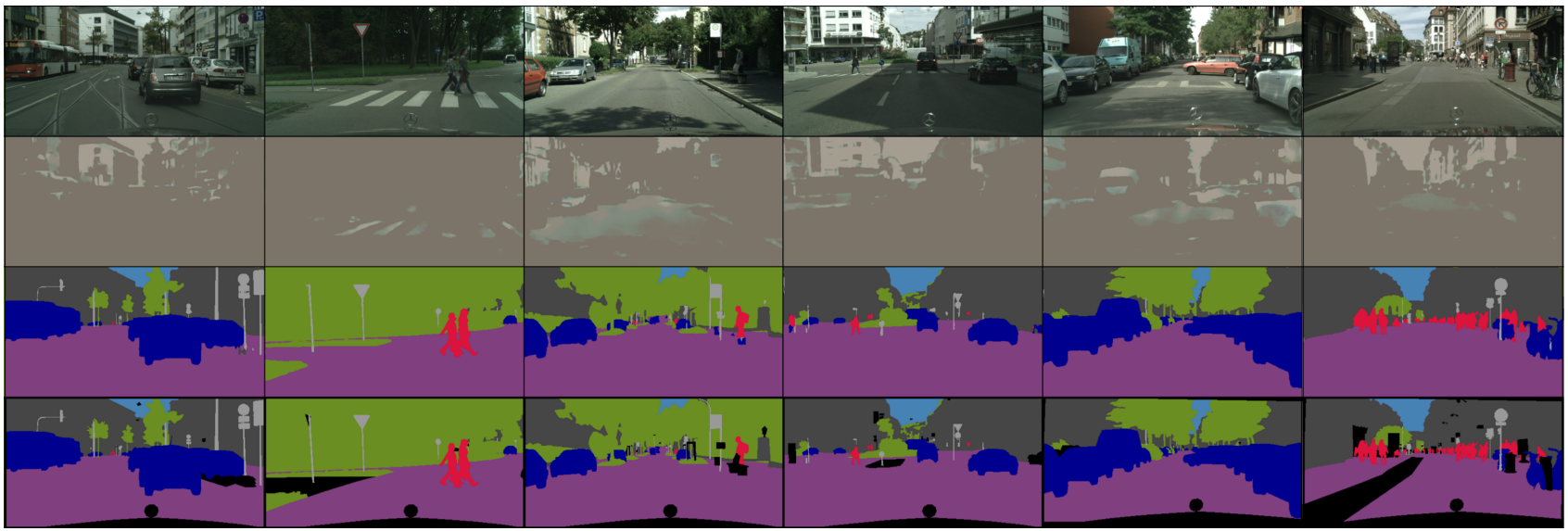


Figure: The first row contains the original image. The second row is filled with the reconstructed images from encoding the original image into z_0 and decoding. The third and fourth rows contain model and ground truth segmentation, respectively.