Compositional Networks

Alan Yuille

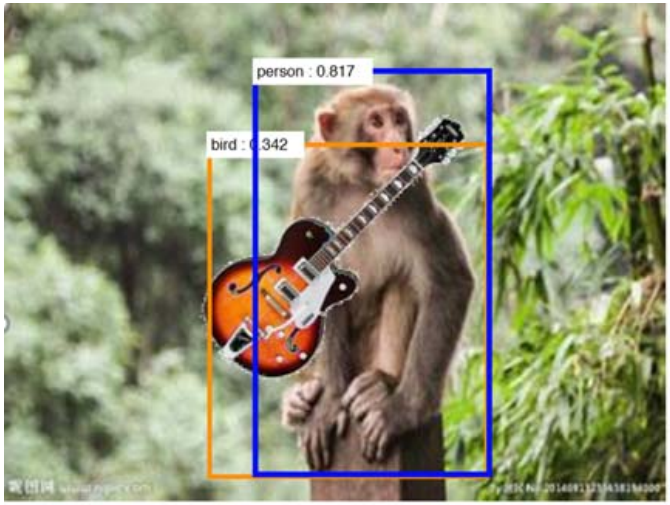
Dept. Cognitive Science and Computer Science

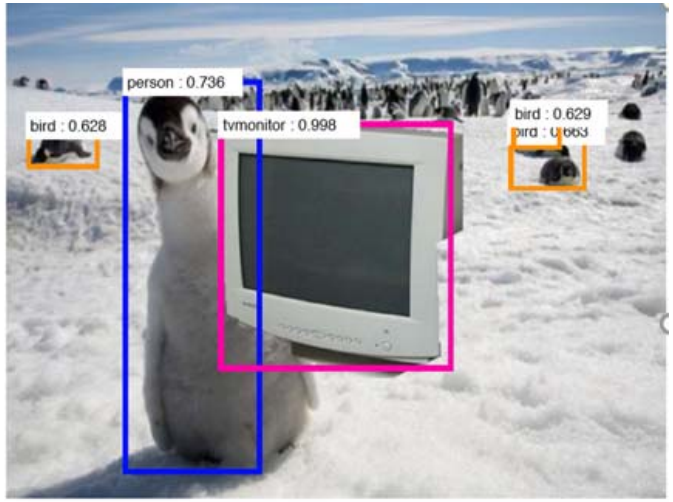
Johns Hopkins University

Background

- Deep Nets are hard to interpret and have unusual failure modes.

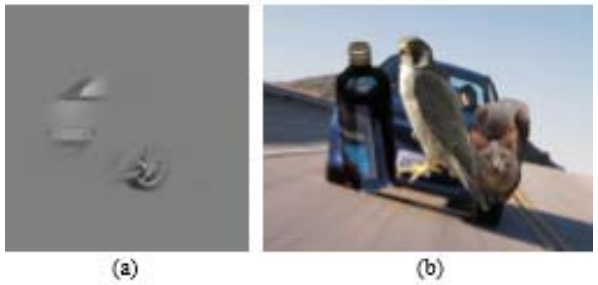
  In particular: they are sensitive to occlusion and context.



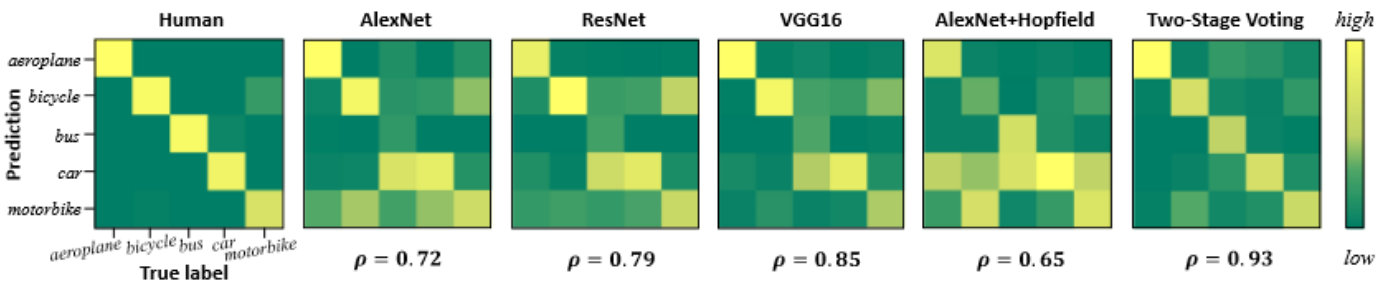


Jianyu Wang, Zhishuai Zhang, Cihang Xie, Yuyin Zhou, Vittal Premachandran, Jun Zhu, Lingxi Xie, and Alan Yuille. Visual concepts and compositional voting. *Annals of Mathematical Sciences and Applications,* 2018.
See also: A Rosenfield et al. The Elephant in the Room. Arxiv. 2018.

How Well Can Humans recognize occluded objects?

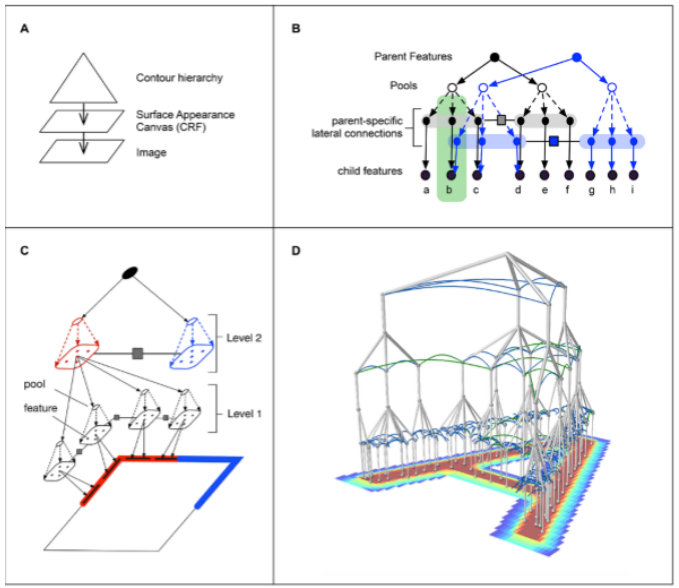


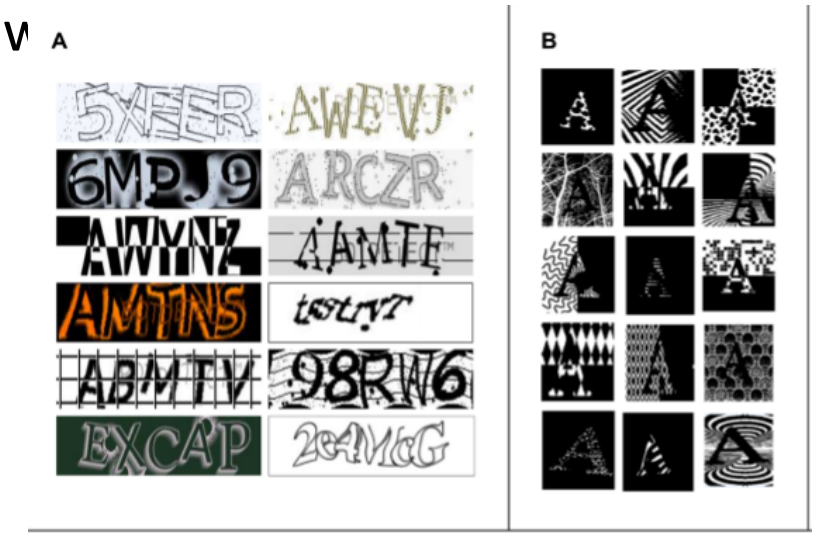
- Hongru Zhu et al. Proc. Cog Sci. 2019.
- Mask occluders. “Extreme” occluders.
-
- Category-level confusion matrices under extreme occlusion. Rho gives the correlation between human and model confusion matrix.



Capcha' D. George et al. 2017.

- A generative vision model that trains w
  breaks text-based CAPTCHAs





Compositional Nets for Object Classification

- *Can Deep Nets be modified to produce better internal representations corresponding to object parts?*
- There has been some work in this direction.
- R. Liao, A. Schwing, R. Zemel, and R. Urtasun. Learning deep parsimonious representations. In Advances in NeurIPSystems. 2016.
- This impose a K-means regularizer on the activity of a layer of neurons. This effectively encourages visual concepts to form. (But re-implementing this made little difference in our case, perhaps because the VCs were already strong).
- An alternative method – maximizing mutual information – gives the ability to detect parts of animals (PascalPart Dataset). Q. Zhang, Y-N. Wu, and S-C Zhu. Interpretable Convolutional Neural Networks. CVPR 2018. (But these are different types of objects and parts than those we are considering).

Deep Networks and Occlusion

- Deep Nets performance degrades on occluded objects.
- *Experiments: Train on un-occluded data and test on Occluded.*
- *Why? Because there are an exponential number of ways to occlude objects. See Neural Architecture talk 28/Oct.*

	zero	one_white	one_noise	one_texture	one_object	five_white	five_noise	five_texture	five_object	nine_white	nine_noise	nine_texture	nine_object	Mean
VGG	99.2	97.9	97.9	97.6	90.3	91.6	90.5	89.7	68.8	54.7	52.3	48.1	47.5	78.9

Different Types of Occluders

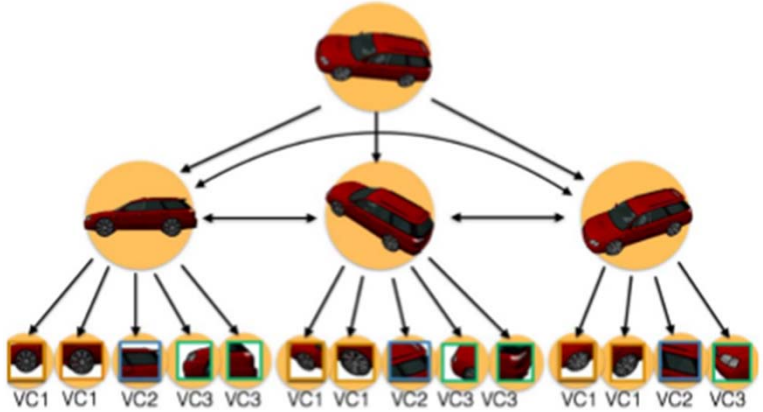
Zero

	White	Noise	Texture	Object
One				
Five				
Nine				

Compositional Nets

- The compositional voting models only work for fixed viewpoint.
- Object appearance depends on the viewpoint. Different VCs will be activated for different viewpoints and in different spatial locations.
- This requires us to use mixture models for objects. *Each mixture component corresponds to a viewpoint and to a spatial pattern of VCs.*
- This must be learned unsupervised

(for fair comparison to Deep Nets).



Two Models: CompNet-Dict & CompNet-Full Hard-VCs and Soft-VCs

- We describe two types of models for each mixture component.
- The models are generative: (i) hard-VCs and (ii) soft-VCs.
- For the hard-VC model, we represent the object by a binary encoding using the VCs.
- We learn a dictionary of VCs as before: $D = \{d_1, \ldots, d_K\}$
- We encode each feature vector f_p by a binary code

$$\tilde{b}_{p,k} = 1 \text{ if } g(f_p, d_k) > \delta.$$

- Empirical finding: each point on the object are encoded by one or two VCs (recall, binary encoding by VCs was mentioned earlier).

CompNet-Dict: Generative Model for Hard-VCs

- For each mixture component A_y^m we learn a Bernoulli distribution for the spatial activation of VCs.

$$p(B|\mathcal{A}_y) = \prod_p p(b_p|\alpha_{p,y}) = \prod_{p,k} \alpha_{p,k,y}^{b_{p,k}} (1 - \alpha_{p,k,y})^{1-b_{p,k}}.$$

- This distribution is factorized (spatially independent). An approximation to simply the model.
- Recall that $\bar{b}_{p,k} = 1$ if VC k is activated at position p.
- The $\alpha_{p,k,y}$ are the parameters of the model (to be leaned).

Occlusion and Robustness

To enable the generative model robust – i.e. able to deal with occlusion – by allowing a probability that the binary encoding is generated by a random background model at some locations.

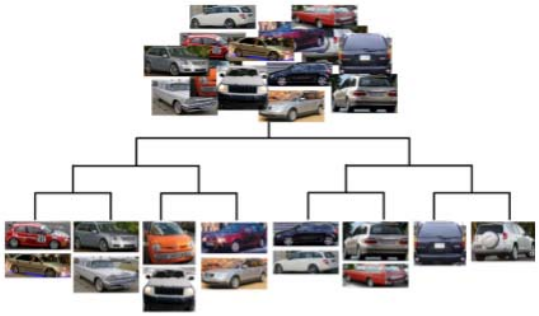
$$
\begin{aligned}
p(B|\Gamma) &= \prod_p p(b_p|FG)^{z_p} p(b_p|BG)^{1-z_p}, \\
z_p &\in \{0,1\}, \\
p(b_p|FG) &= p(b_p|\alpha_{p,y}) p(z_p), \\
p(b_p|BG) &= p(b_p|\beta)(1-p(z_p)).
\end{aligned}
$$

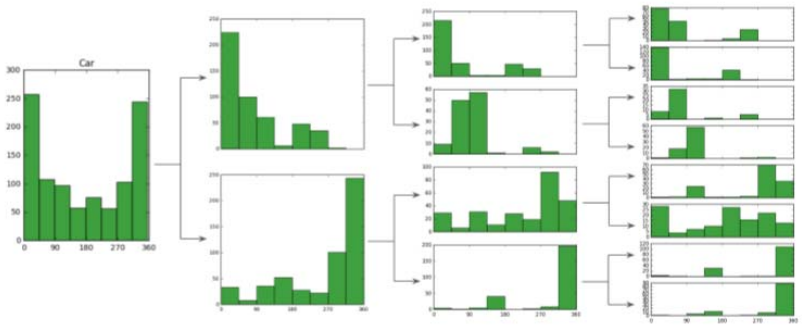
The mixture models.

- An object is represented by a mixture of distributions:

$$p(B|\mathcal{A}_y, \mathcal{V}) = \prod_m p(B|\mathcal{A}_y^m)^{\nu_m}, \sum_m \nu_m = 1,\ \nu_m \in \{0,1\}.$$

- This model can be learnt by the EM algorithm. The number of mixture components for each object is learnt automatically by clustering. The intuition is that mixture components have similar VC spatial patterns





CompNet-Full.
Generative Model for Hard-VC encoding

- Generative models are learnt for all objects. The only supervision is object identity. The learning algorithm involves backprop, clustering, and EM.
- CompNet-Full is much more effective that standard deep networks if there is significant occlusion. Explicit representation in terms of parts allows them to be switched off automatically if there is an occluder (hard to do for a Deep Net with only explicit representations).
- But this model is not effective at localizing occluders, despite being robust to them. (Results will be shown later).
- A. Kortylewski et al. In submission. 2019.

CompNet-Full
A Generative Model for Soft-VCs

- We define a second generative model – CompNet-Full, which also represents objects in terms of mixtures of spatial patterns of VCs.
- Now the mixture components are defined over the feature vectors using soft-VC encoding. This is more robust than hard-encoding.
- This replaces the Bernoulli distribution over the binary-encodings (four slides previously) by a von Mises-Fisher mixture distribution over the feature vectors, where each mixture component corresponds to a VC.
- Recall that we could learn the VCs by using von Mises-Fisher distributions to cluster them. (Note: von Mises-Fisher is analogous to mixtures of Gaussians but for normalized feature vectors).

Von Mises-Fisher Distribution

- We replace the Bernoulli distribution by a distribution over the feature vectors:

$$p(F|\Theta_y) = \prod_p p(f_p|\mathcal{A}_{p,y}, \theta) = \prod_p \sum_k \alpha_{p,k,y} p(f_p|S_k, \mu_k), \tag{5}$$

where $\Theta_y = \{\mathcal{A}_{0,y}, \ldots, \mathcal{A}_{\mathcal{P},y}, \theta\}$ are the model parameters at every position $p \in \mathcal{P}$ on the lattice of the feature map F, $\mathcal{A}_{p,y} = \{\alpha_{p,0,y}, \ldots, \alpha_{p,K,y} | \sum_{k=0}^{K} \alpha_{p,k,y} = 1\}$ are the mixture coefficients, K is the number of mixture components, $\theta = \{\theta_k = \{S_k, \mu_k\} | k = 1, \ldots, K\}$ are the parameters of the vMF mixture distributions:

$$p(f_p|S_k, \mu_k) = \frac{e^{S_k \mu_k^T f_p}}{Z(S_k)}, \|f_p\| = 1, \|\mu_k\| = 1, \tag{6}$$

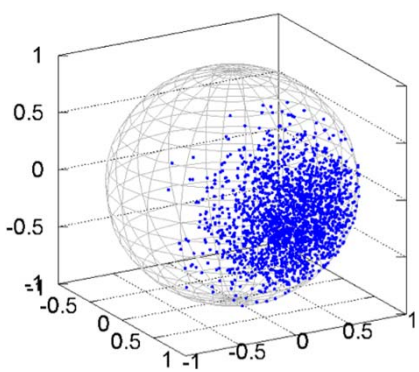
- Z is a normalizing constant.

Von Mises-Fisher versus Bernoulli

Bernoulli Distribution

$$p(B|\mathcal{A}_y) = \prod_p p(b_p|\alpha_{p,y}) = \prod_{p,k} \alpha_{p,k,y}^{b_{p,k}} (1 - \alpha_{p,k,y})^{1-b_{p,k}}$$

Von Mises-Fisher Distribution. Feature vectors normalized to lie on unit sphere.



$$p(F|\Theta_y) = \prod_p p(f_p|\mathcal{A}_{p,y}, \theta)$$
$$= \prod_p \sum_k \alpha_{p,k,y} p(f_p|S_k, \mu_k),$$

$$p(f_p|S_k, \mu_k) = \frac{e^{S_k \mu_k^T f_p}}{Z(S_k)}, \|f_p\| = 1, \|\mu_k\| = 1,$$

Mixtures and Robustness to Occluders

- Objects are represented by a mixture of distributions, where each mixture is a factorized product of Fisher von-Mises distributions over the input feature vectors.
- We make this model robust using the same mechanism as before, which allows some feature vectors to be generated randomly.

$$p(F|\Theta_y, \beta) = \prod_p [p(f_p|FG)p(z_p)]^{z_p} [p(f_p|BG)p(1-z_p)]^{1-z_p} \qquad z_p \in \{0,1\}$$

$$p(f_p|FG) = \sum_k \alpha_{p,k,y} p(f_p|S_k, \mu_k)$$

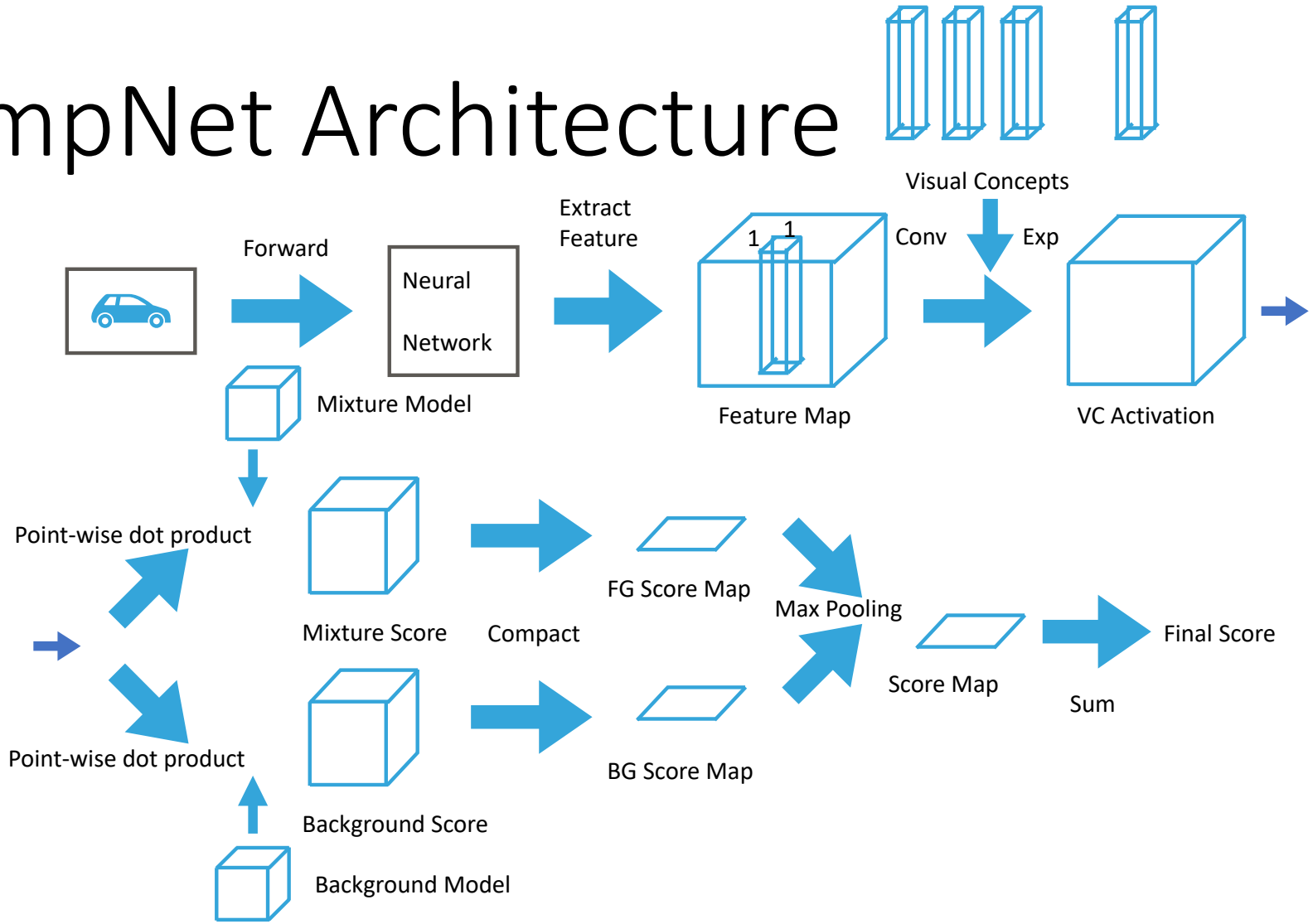
$$p(f_p|BG) = \sum_k \beta_k p(f_p|S_k, \mu_k)$$

CompNet-Full

- The parameters of this model, and the number of mixture components, are learnt automatically. Clustering is used to estimate the number of mixtures (and to group the training data into them).
- This initializes an EM algorithm which learns all the parameters.
- The only supervision is the name of the object.

- (The model can be trained end-to-end, but this is beyond the scope of this talk).

CompNet Architecture



Compare CompNet-Dict, CompNet-Full, and Deep Net (VGG) on the Occlusion Dataset

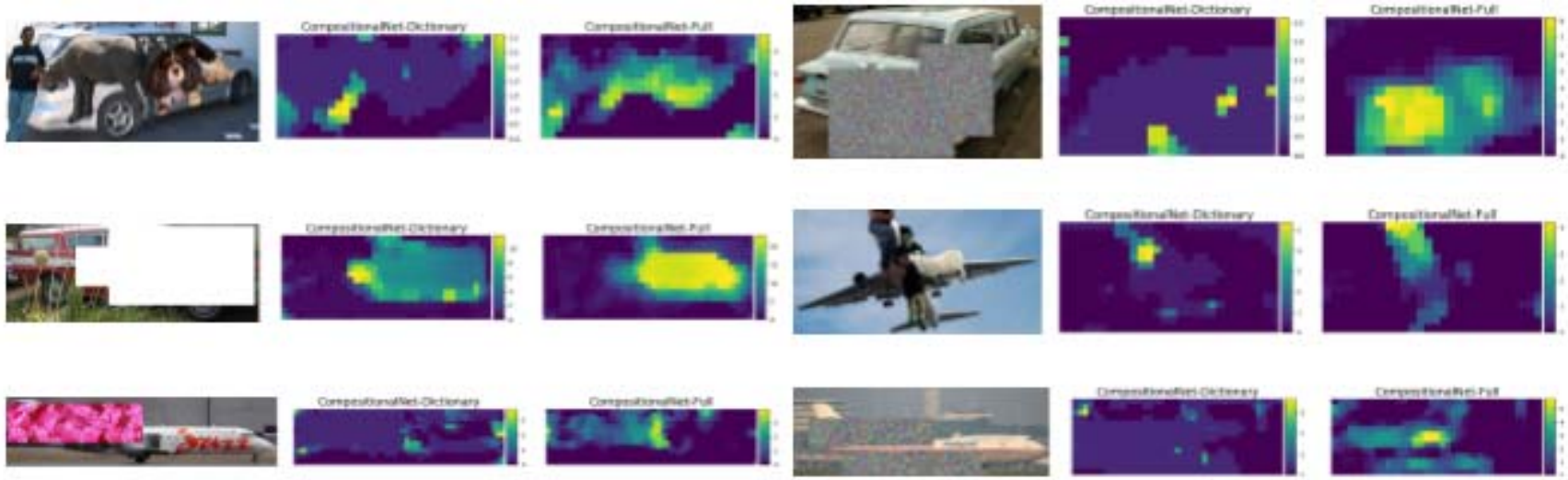
- Both CompNet models do than Deep Nets as the Occlusion increases.
- CompNet-Full (soft-VCs) slightly outperforms CompNet-Dict (hard VCs), but both significantly outperform VGG as occlusion increases.

Classification under Occlusion

Occ. Area	0%	Level-1: 20-40%				Level-2: 40-60%				Level-3: 60-80%				Mean
Occ. Type	-	w	n	t	o	w	n	t	o	w	n	t	o	-
VGG	99.2	97.9	97.9	97.6	90.3	91.6	90.5	89.7	68.8	54.7	52.3	48.1	47.5	78.9
CompMixOcc-Dict	92.1	92.7	92.3	91.7	92.3	87.4	89.5	88.7	90.6	70.2	80.3	76.9	87.1	87.1
CompMixOcc-Full	95.9	95.8	95.2	94.9	94.9	95.0	93.3	92.9	92.3	86.8	83.8	80.9	88.1	91.5
CompNet-Dict	98.3	96.8	95.9	96.2	94.4	91.2	91.8	91.3	91.4	71.6	80.7	77.3	87.2	89.5
CompNet-Full	98.6	97.9	97.5	97.3	96.1	95.9	94.5	94.1	92.4	86.8	84.0	80.9	87.7	92.6
Human	100.0	100.0				100.0				98.3				99.5

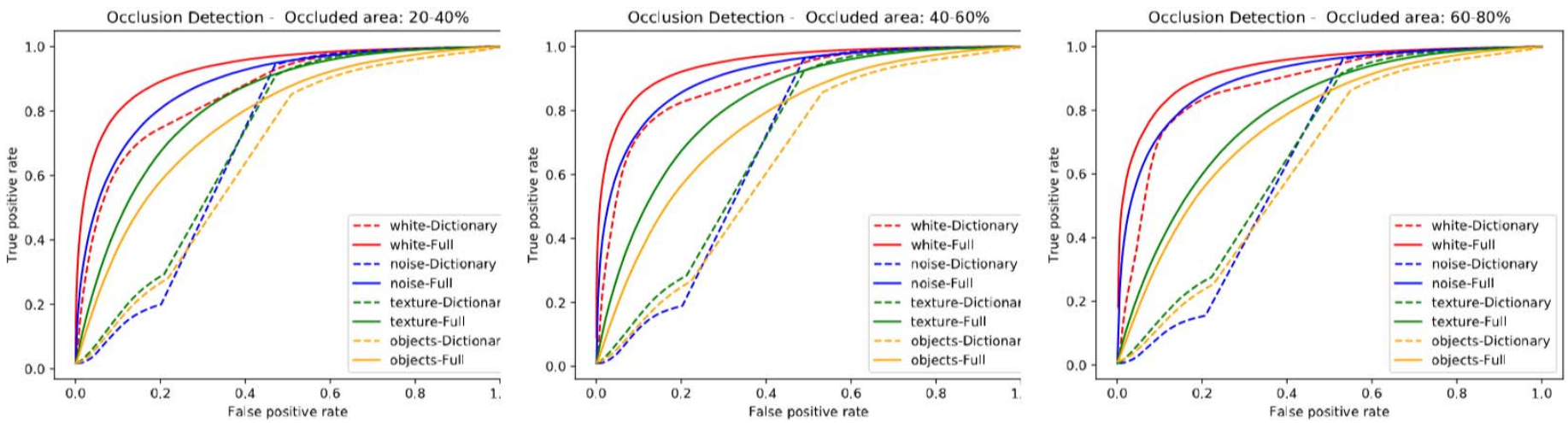
CompNet-Full: Detect and Localize Occluders

- The CompNets can detect and locate occludes by determining where the model uses robustness (i.e. where the input feature vectors are significantly different than those predicted by the model).
- Here are some visual examples (not cherry picked).
- Left: Image. Center: CompNet-Dict. Right: CompNet-Full



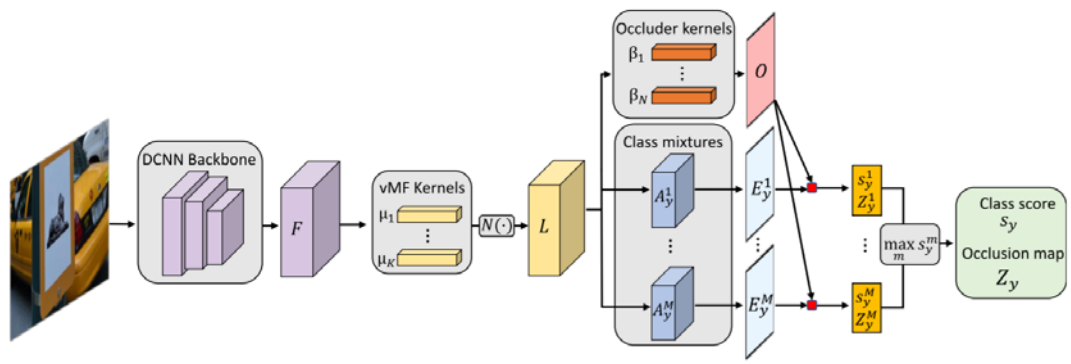
Compare CompNets to detect/localize occluders

- CompNet-Full (solid lines) outperforms CompNet-Dict (dashed lines) to detect/localize occluders, for all types (White, Noise, Texture, Objects).
- Left to Right: Occlusion Levels 20-40%, 40-60 %, 60-80 %.



End-To-End Model

- These compositional deep network models are variants of deep networks which combine standard backpropagation with clustering.
- These compositional deep networks have the advantages of deep networks (e.g., end-to-end training) and of compositional models (explicit parts).
- Caveat – we develop these models for six classes of vehicles only.



The Loss Function

- This is composed of four terms:

$$\mathcal{L}(y, y', F, T) = \mathcal{L}_{class}(y, y') + \gamma_1 \mathcal{L}_{weight}(\omega) + \gamma_2 \mathcal{L}_{vmf}(F, \Lambda) + \gamma_3 \mathcal{L}_{mix}(F, \mathcal{A}_y).$$

- The first two terms are the cross-entropy loss and a weight regularizer.
- The last two terms enforces the VCs and the mixtures.

$$\mathcal{L}_{vmf}(F, \Lambda) = -\sum_p \max_k \log p(f_p|\mu_k) = C \sum_p \min_k \mu_k^T f_p,$$

$$\mathcal{L}_{mix}(F, \mathcal{A}_y) = -\sum_p (1 - z_p^{\dagger}) \log \Big[\sum_k \alpha_{p,k,y}^{m^{\dagger}} p(f_p|\lambda_k) \Big]$$

Performance Comparision: Pascal3D+ and Coco

- We have two datasets: (1) Pascal3D+ (left) where occlusion is photoshopped, (2) Coco (right) with annotated occlusion-level.
- Compositional Nets outperform Deep Networks (slightly) on non-occluded datasets.
- CNets outperform DNets and significantly (enormously) as the amount of occlusion increases.

Results: Object Classification

- PASCAL (left). Coco (bottom: right).Occlusuon localization (right).

PASCAL3D+ Vehicles Classification under Occlusion

Occ. Area	L0: 0%	L1: 20-40%				L2: 40-60%				L3: 60-80%				Mean
Occ. Type	-	w	n	t	o	w	n	t	o	w	n	t	o	
VGG	99.2	96.9	97.0	96.5	93.8	92.0	90.3	89.9	79.6	67.9	62.1	59.5	62.2	83.6
CoD[11]	92.1	92.7	92.3	91.7	92.3	87.4	89.5	88.7	90.6	70.2	80.3	76.9	87.1	87.1
VGG+CoD [11]	98.3	96.8	95.9	96.2	94.4	91.2	91.8	91.3	91.4	71.6	80.7	77.3	87.2	89.5
CompNet-p4	97.4	96.7	96.0	95.9	95.5	95.8	94.3	93.8	92.5	86.3	84.4	82.1	88.1	92.2
CompNet-p5	**99.3**	98.4	**98.6**	98.4	96.9	98.2	98.3	97.3	88.1	90.1	89.1	83.0	72.8	93.0
CompNet-Multi	**99.3**	**98.6**	**98.6**	**98.8**	**97.9**	**98.4**	**98.4**	**97.8**	**94.6**	**91.7**	**90.7**	**86.7**	**88.4**	**95.4**

MS-COCO Vehicles Classification under Occlusion

Train Data	PASCAL3D+					MS-COCO					MS-COCO + CutOut					MS-COCO + CutPaste				
Occ. Area	L0	L1	L2	L3	Avg	L0	L1	L2	L3	Avg	L0	L1	L2	L3	Avg	L0	L1	L2	L3	Avg
VGG	97.8	86.8	79.1	60.3	81.0	99.1	88.7	78.8	63.0	82.4	99.3	90.9	87.5	75.3	88.3	99.3	92.3	89.9	80.8	90.6
CompNet-p4	96.9	91.8	82.9	70.1	87.1	97.1	95.5	89.9	85.5	89.1	97.8	91.9	87.9	75.2	89.5	98.3	93.8	88.9	84.9	91.4
CompNet-p5	98.5	98.1	94.3	87.1	93.2	99.1	95.2	87.3	85.5	90.3	99.3	95.5	87.9	84.9	91.6	**99.4**	93.9	90.0	81.8	93.2
CompNet-Multi	**98.2**	**98.8**	**98.9**	**95.5**	**97.9**	**99.4**	**95.3**	**90.0**	**89.3**	**93.0**	**99.4**	**95.5**	**90.2**	**89.3**	**95.6**	**99.4**	**95.8**	**91.8**	**88.4**	**94.4**

Summary

- CompNet architectures that could classify vehicles with significant occlusion. These models were interpretable. They could detect/localize occluders, localize subparts (VCs).

- End-to-end training. Outperforming deep networks on Pascal3D+ and Coco. With and without occlusion.