Deep Compositional Networks

Alan Yuille

Dept. Cognitive Science and Computer Science

Johns Hopkins University

Abstract

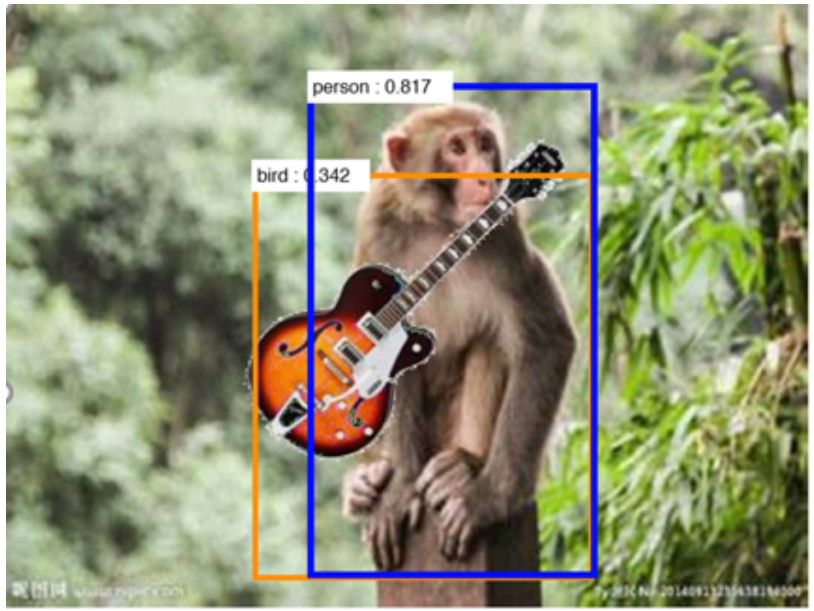
- This talk describes a family of compositional networks which are interpretable and perform tasks like object classification and part recognition.
- Moreover, they outperform conventional deep networks in challenging situations where there is extreme occlusion.

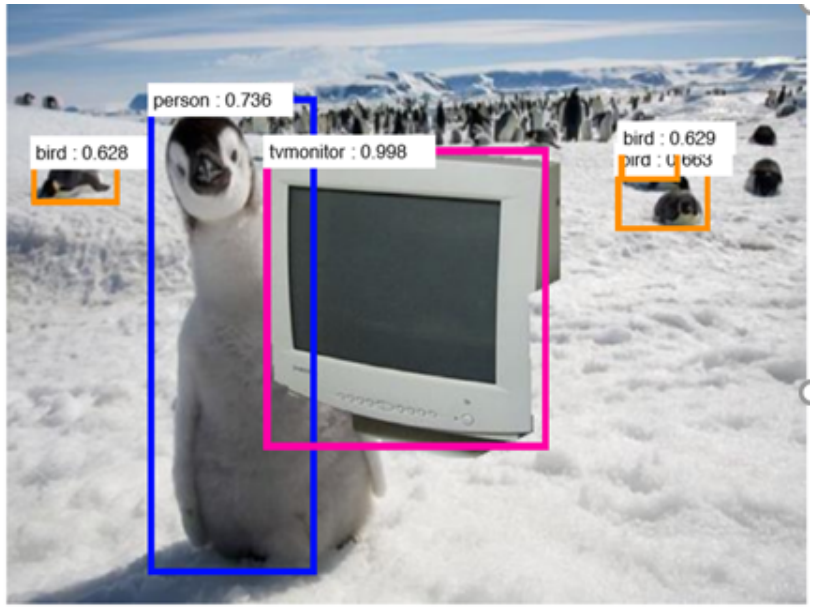
- *For more details: see poster by A. Kortylewski et al. Localizing Occluders with Compositional Convolutional Networks. Neural Architecture Workshop. ICCV 28/Oct.*

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.

PART 1: Visual Concepts: Internal Representations

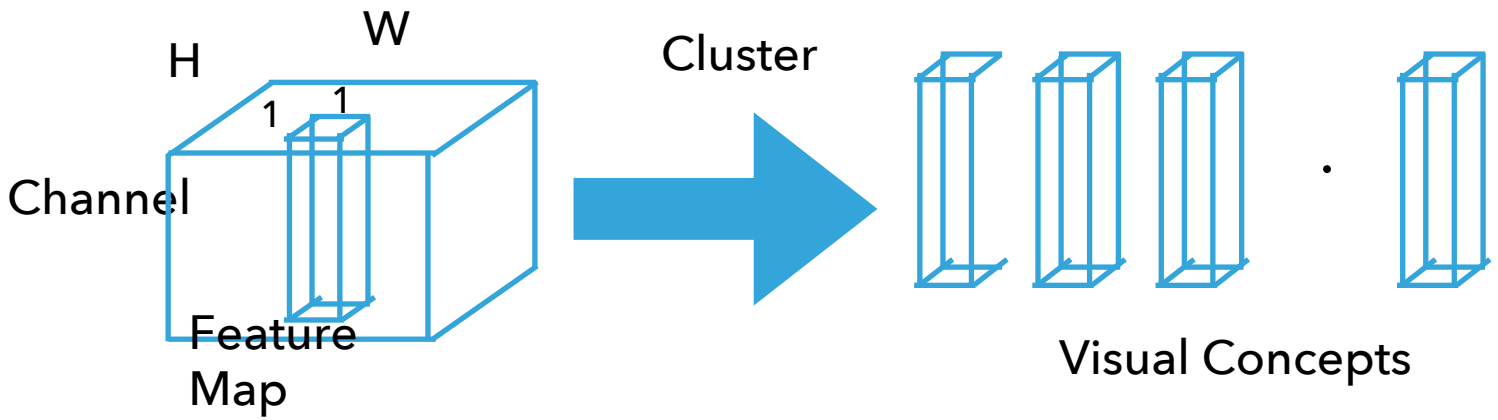
- We study internal representations within Deep Nets.
- We restrict ourselves to study vehicles at fixed scale from the Pascal3D+ dataset.
- We showed that visual concepts, encoded by feature populations, represented subparts of the vehicles.
- We quantified the visual concepts for a series of tasks including semantic part detection under occlusion.
- *J. Wang et al. Visual concepts and compositional voting. Annals of Mathematical Sciences and Applications, 2018.*
- *J. Wang et al. Detecting Semantic Parts on Partly Occluded Objects. BMVC. 2017.*

Background

- It has been shown (e.g., B. Zhou et al. ICLR 2015) that deep nets contain internal representations represented by neural features. The findings included:
- (I) If Deep Nets are trained to perform scene recognition, then the internal representations correspond to objects.
- (II) If Deep Nets are trained to perform object recognition, then the internal representations correspond to object parts.
- *For related work, see A. Vedaldi's presentation in this tutorial session*.

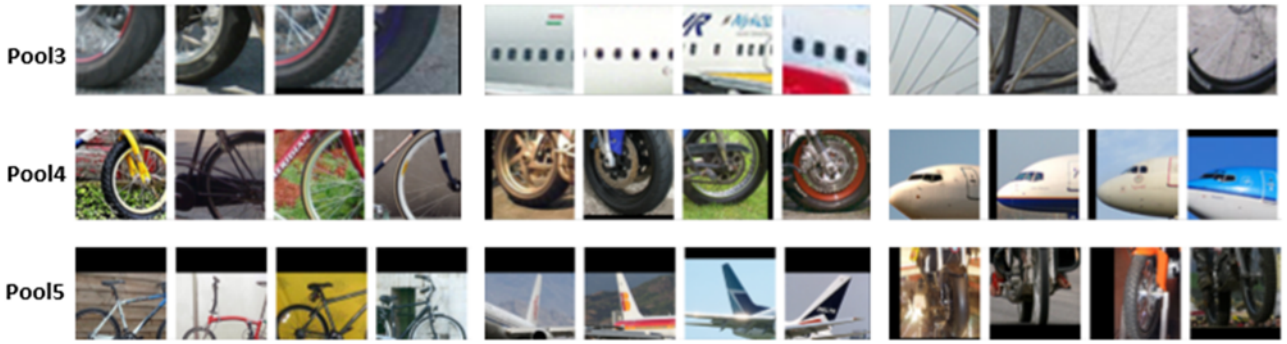
Visual Concepts

- We conjectured that subparts of objects are encoded by populations of feature vectors – instead of by features themselves.
- These *visual concepts* were found by clustering the feature vectors. We restricted ourselves to vehicles from Pascal3D+ and fixed the scale of the objects.



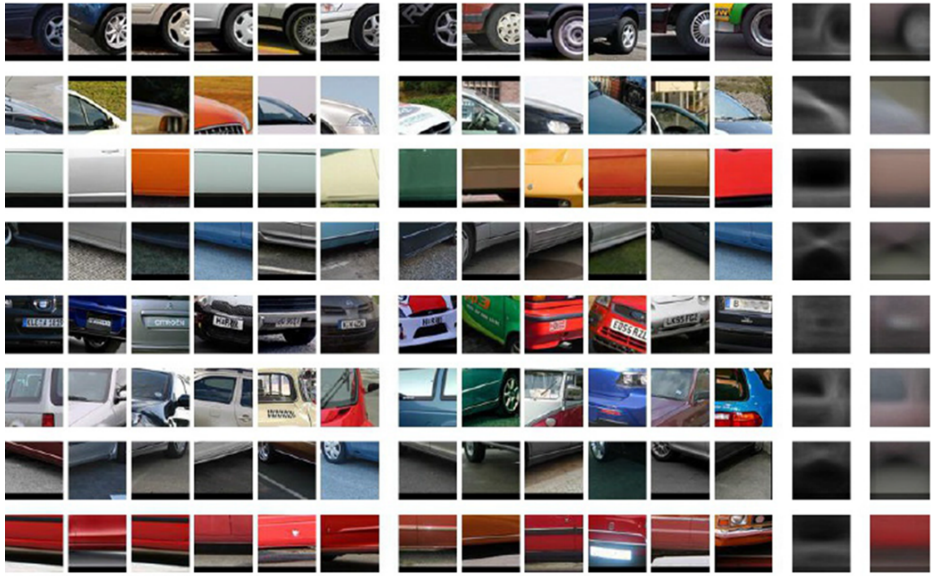
Visual Concepts: Clustering

- The clustering was done using k-means with k=200 (alternative clustering methods, and alternative values of k gave similar results).
- The clustering was done at different levels of the Deep Net. E.g., Pool3, Pool4, Pool5. Results were similar for AlexNet and VGG.
- Visual Concepts correspond to parts of objects. VCs at higher layers correspond to larger parts (e.g., Pool4 wheel, Pool3 wheel-part).



Visual Concepts: Perceptually Tight

- *Findings 1: The visual concepts were perceptually tight. Image patches corresponding to the same visual concept are very similar.*
- We show the closest 6 image patches (left), *a random sample of 6 patches from the top 500 image patches* (center), and *the mean of the edge map and of the patches of the top 500 patches* (right).



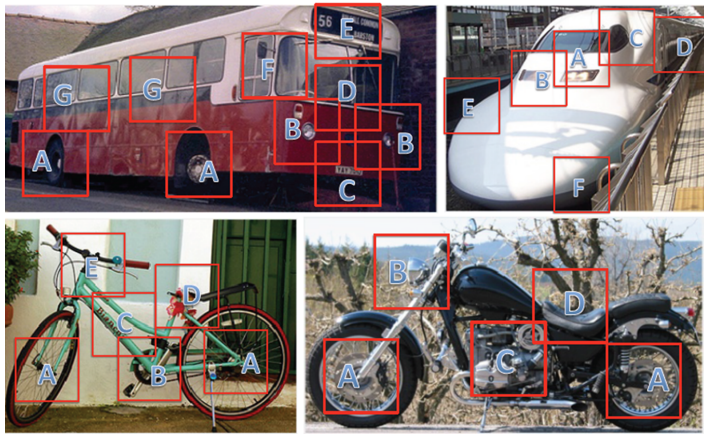
Visual Concepts: Coverage of the Object

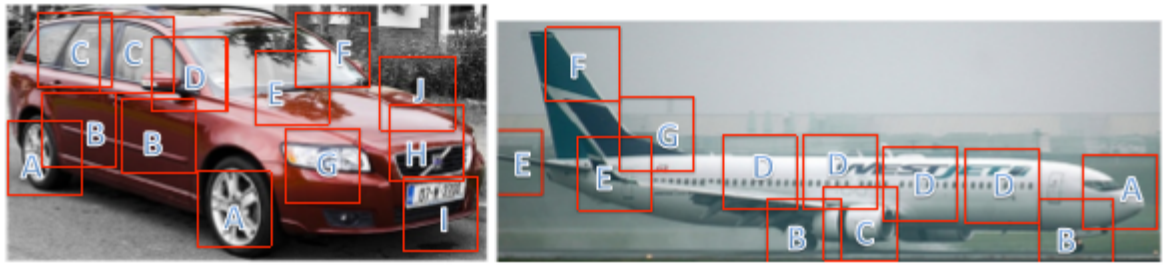
- *Visual Concepts respond to (cover) almost all parts of the object.*
- Here are 44 (out of 170) VCs for cars.
- This can be quantified, by showing that the objects could be represented in terms of VCs by binary encoding (see later).

To Explore: We Annotate Semantic Parts.

- We annotated the vehicles in PASCAL 3D+.

To create the *Vehicle Semantic Part dataset.*





VCs as Key-Point, Semantic Part Detectors.

- VCs were fairly good for detecting key-points and semantic parts of the Vehicles. But much worse than supervised models.

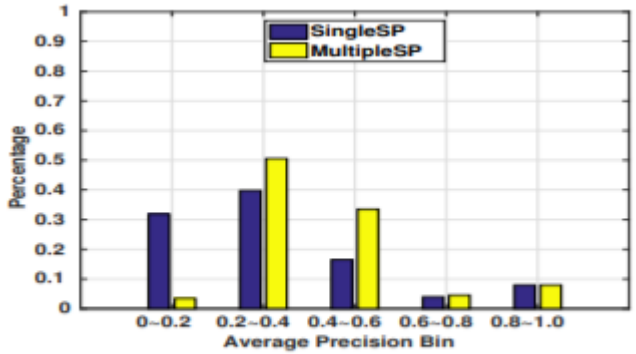
- Key-Points.

13 K-Ps for Bike.

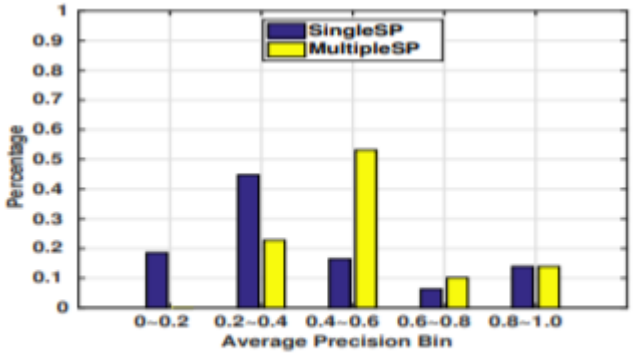
Bike	1	2	3	4	5	6	7	8	9	10	11	12	13	mAP
SF	.77	.84	.89	.91	.94	.92	.94	.91	.91	.56	.53	.15	.40	.75
VC	.91	.95	.98	.96	.96	.96	.97	.96	.97	.73	.69	.19	.50	.83

- Semantic Parts.

Yellow bars show the best APs for each VC.



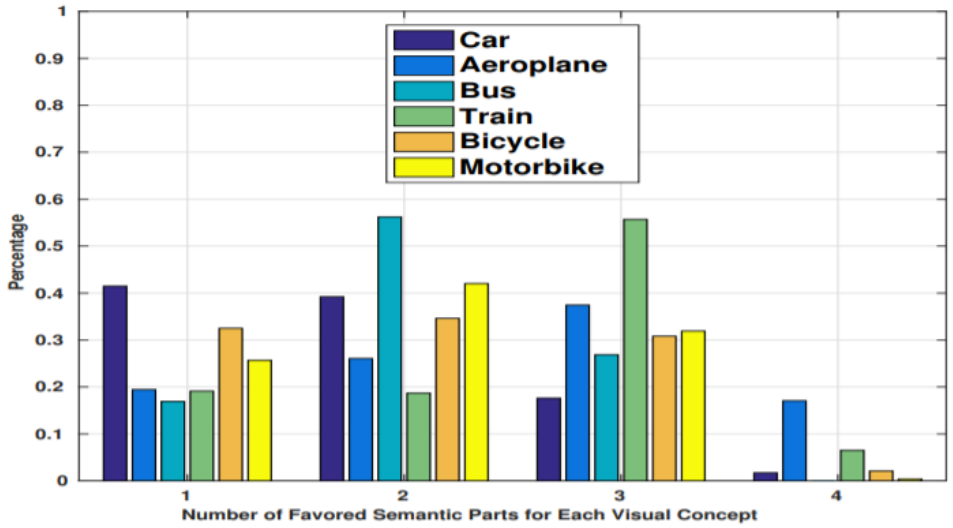
(a) car

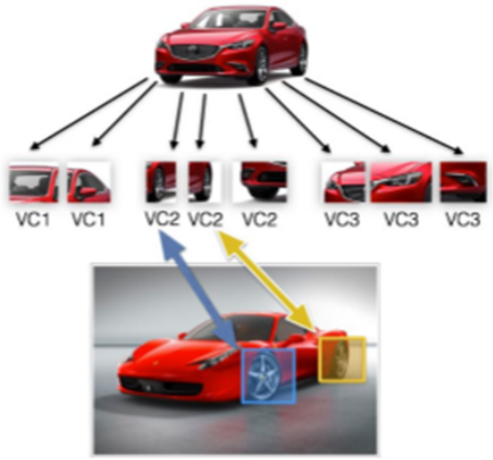


(e) bicycle

VCs detect subparts of Semantic Parts

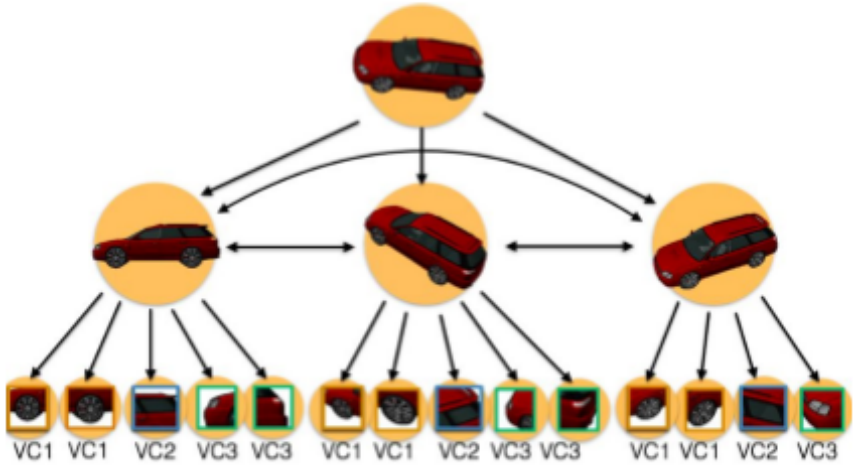
- VCs can act as unsupervised detectors for key-points and semantic-parts. Their Average Precisions (APs) are weaker than supervised methods.
- We observe that most VCs respond to several different semantic parts (typically 1-4). The VCs correspond to subparts of semantic parts (which are shared).



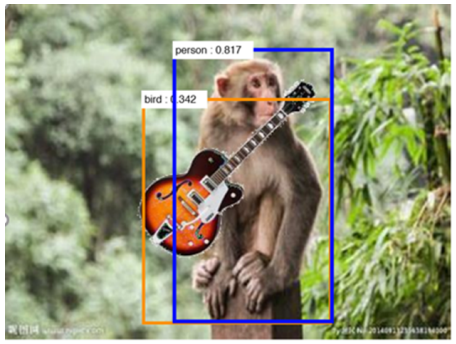


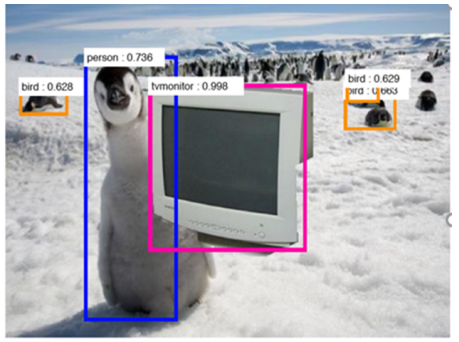
Combine VCs to detect Semantic Parts

- We design a compositional model for detecting semantic parts. Each model consists of a set of VCs which fire in different spatial positions. (Illustrated for object – car – instead of semantic part).
- Compositional Voting: each VC votes for the semantic part (depending on spatial position).



Semantic Part Detection with Occlusion





- We introduce occlusion to make semantic part detection more challenging. *Vehicle Occlusion Dataset.*
- Our intuition is that Deep Nets have difficulty with occlusion. *But compositional voting is likely to be most robust*. The occluded VC will not respond, but the un-occluded VCs will still vote.
- *Compositional voting* also includes context, image information outside the semantic part, because this is also robust.

Detecting Semantic Parts with Occlusion

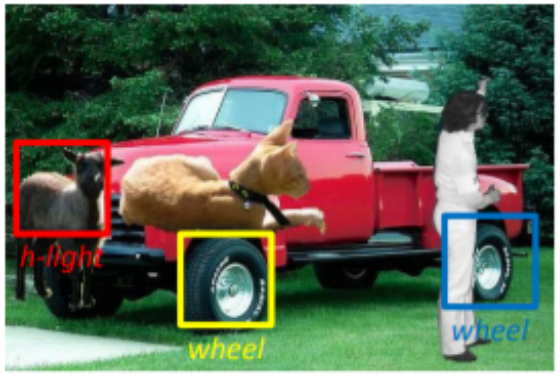
- In the occlusion dataset semantic parts

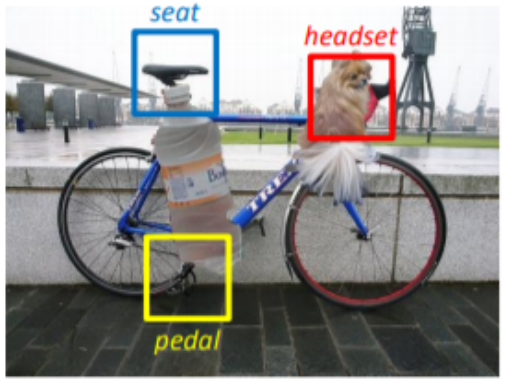
  can be: (i) fully occluded (red)

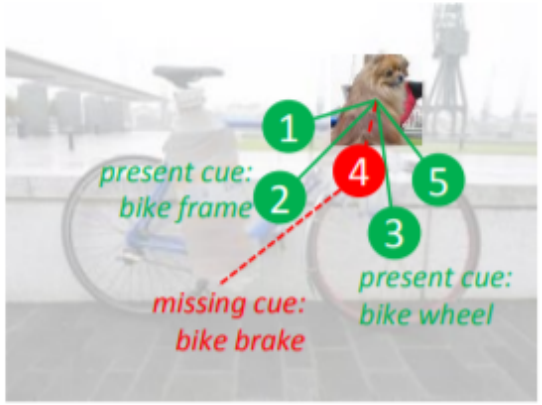
  (ii) partially occluded (blue)

  (iii) un-occluded (yellow).

- Compositional voting uses VCs on and off the semantic parts. If a VC is detected (green) then it votes for the semantic part. If a VC is occluded (red) then it gives no vote.
- Note: a semantic part can be detected even if it is fully occluded.







Compositional Voting: Detect Semantic Parts

- The compositional voting method (VT) outperforms alternatives like Deep Nets if there is significant occlusion.
- *Main idea: explicit representation of subparts (by VC) enables the algorithm to switch them on and off automatically. This makes them robust to occlusion.*

Object	2 Occ's, $0.2 \leqslant r < 0.4$			3 Occ's, $0.4 \leqslant r < 0.6$			4 Occ's, $0.6 \leqslant r < 0.8$		
	SV	**FR**	**VT**	**SV**	**FR**	**VT**	**SV**	**FR**	**VT**
airplane	12.0	**26.8**	23.2	9.7	**20.5**	19.3	7.5	**15.8**	15.1
bicycle	44.6	65.7	**71.7**	33.7	54.2	**66.3**	15.6	37.7	**54.3**
bus	12.3	**41.3**	31.3	7.3	**32.5**	19.3	3.6	**21.4**	9.5
car	13.4	**35.9**	**35.9**	7.7	22.0	**23.6**	4.5	**14.2**	13.8
motorbike	11.4	35.9	**44.1**	7.9	28.8	**34.7**	5.0	19.1	**24.1**
train	4.6	20.0	**21.7**	3.4	**11.1**	8.4	2.0	**7.2**	3.7
mean	16.4	37.6	**38.0**	11.6	28.2	**28.6**	6.4	19.2	**20.1**

- *J. Wang et al. BMVC (2017). See also, Z. Zhang et al. CVPR. 2018.*

Visual Concepts: Summary

- The Deep Nets encode representations of the parts. These are stored by the activity patterns of the feature vectors (individual features were less successful – quantitatively). *Note: vehicles only (rigid classes) and fixed scale.*
- *Making this representation explicit – e.g., by compositional voting – enables us to detect semantic parts despite heavy occlusion.* The algorithm can automatically switch off subparts (VCs) if they are not detected in the correct locations.
- It is harder for Deep Nets to deal with occluders, because their representations are not explicit, so it is difficult to switch parts off.
- *Can we extend this too classify objects?*

Part 2: 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

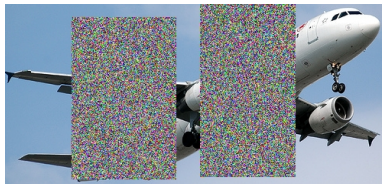
- Deep Nets performance degrades on occluded objects.
- *Experiments: Train on un-occluded data and test on Occluded.*
- *(Why? Because there are an exponential number of ways to occlude objects. See Neural Architecture talk 28/Oct).*




	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

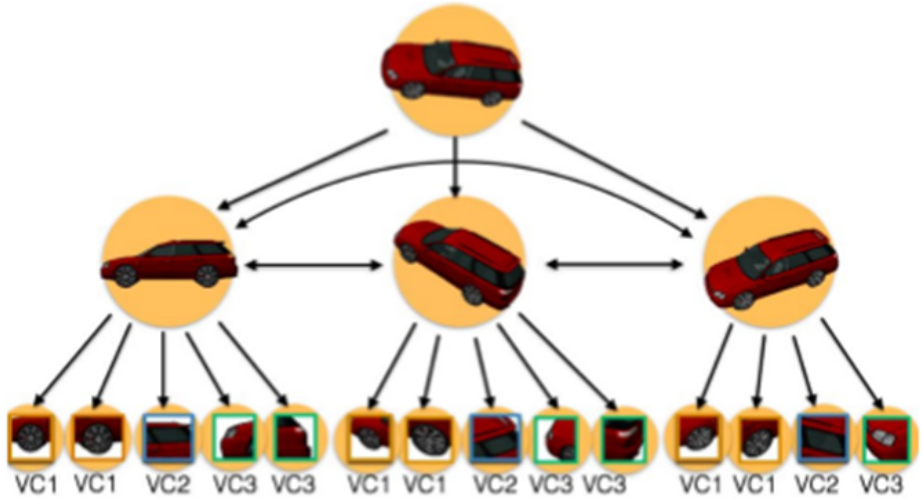
Zero

	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

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

- 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.

$$p(B|\Gamma) = \prod_p p(b_p|FG)^{z_p} p(b_p|BG)^{1-z_p},$$

$$z_p \in \{0, 1\},$$

$$p(b_p|FG) = p(b_p|\alpha_{p,y})p(z_p),$$

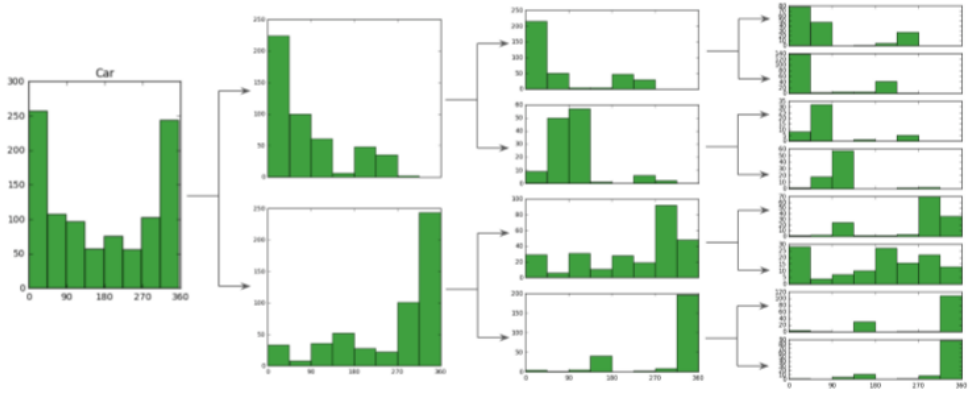
$$p(b_p|BG) = p(b_p|\beta)(1 - p(z_p)).$$

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:

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

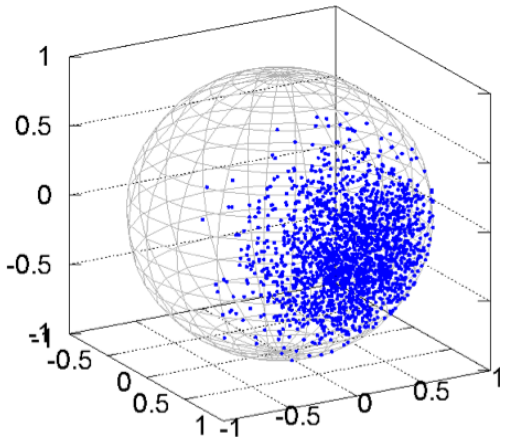
- Z is a normalizing constant.

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.

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

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

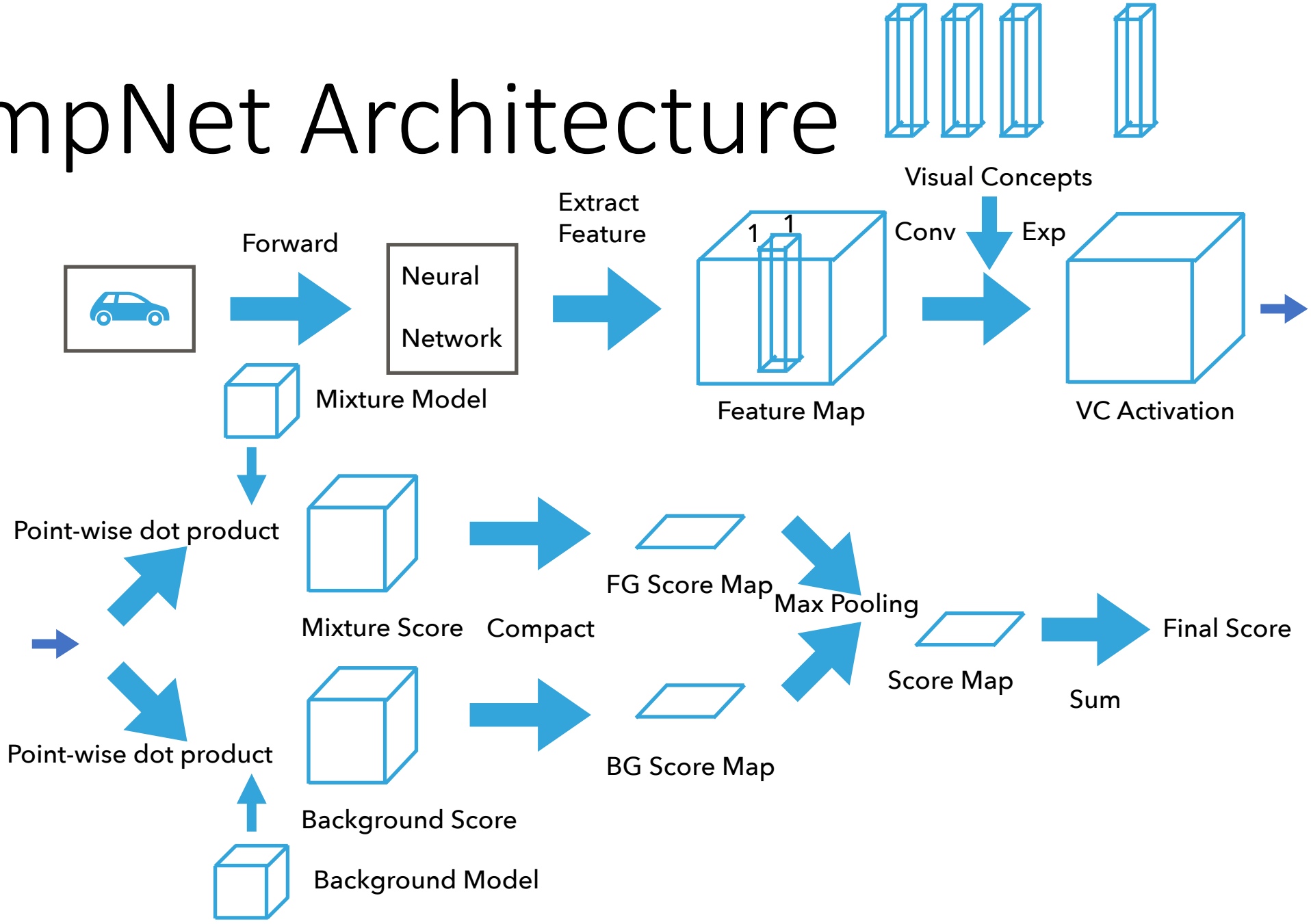
$$p(f_p|BG) = \sum_k \beta_k p(f_p|S_k, \mu_k)$$

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

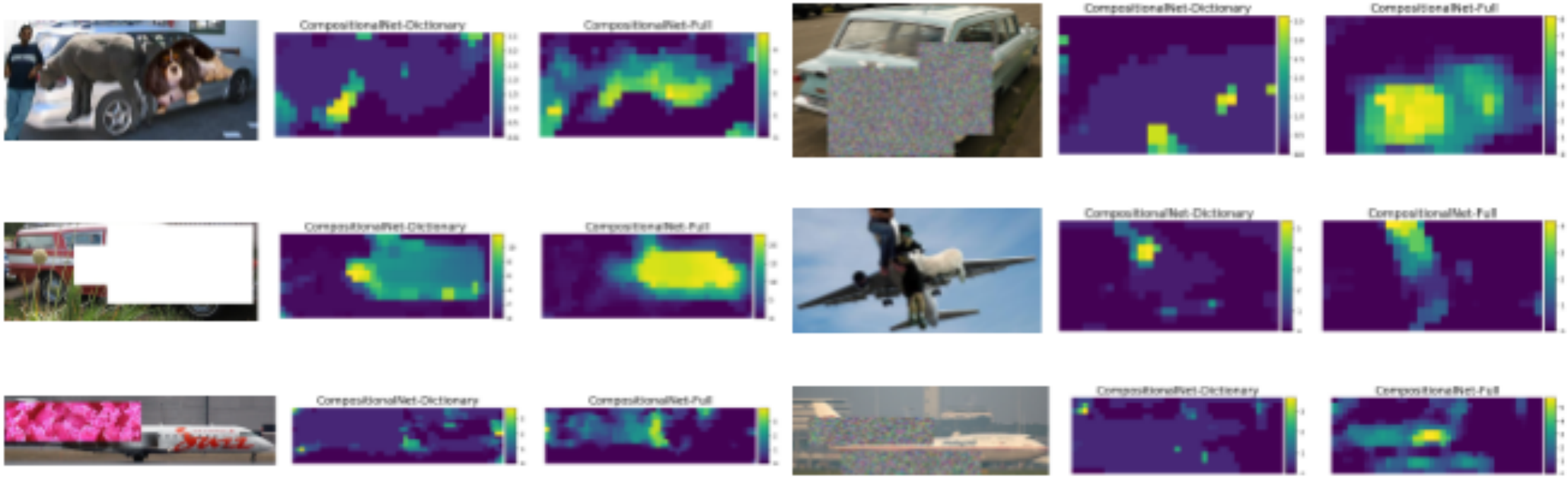
- Both CompNet models do than Deep Nets as the Occlusion increases.
- CompNet-Full (soft-VCs) slightly outperforms CompNet-Dict (hard VCs), but both significantly outperform VGG as occlusion increases.

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

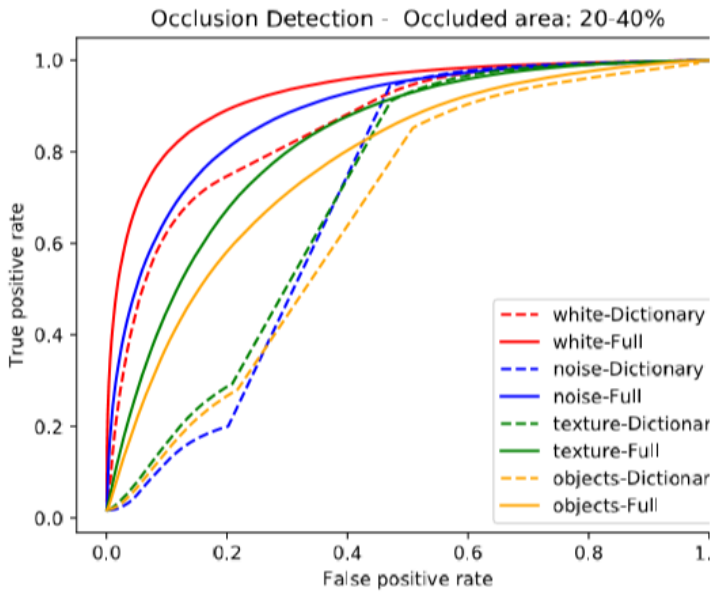
CompNet-Full: Detect and Localize Occluders

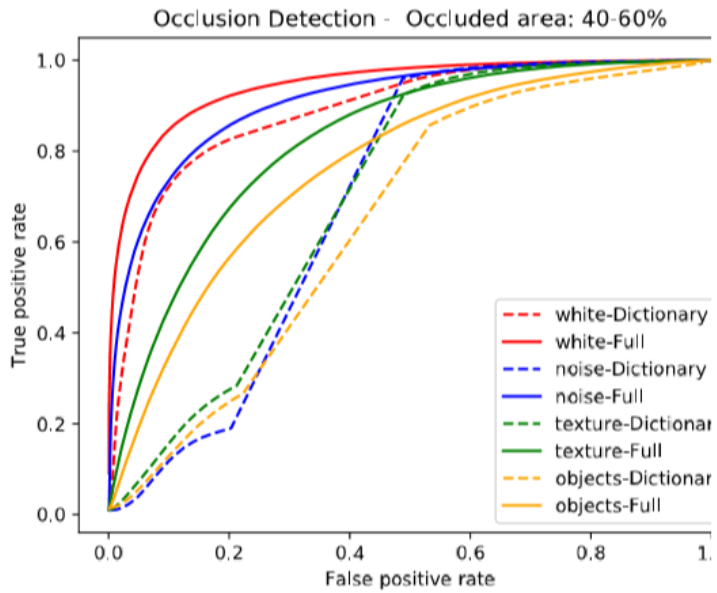
- The CompNets can detect and locate occludes by determining where the model uses robustness (i.e. where the input feature vectors are significantly different than those predicted by the model).
- Here are some visual examples (not cherry picked).
- Left: Image. Center: CompNet-Dict. Right: CompNet-Full

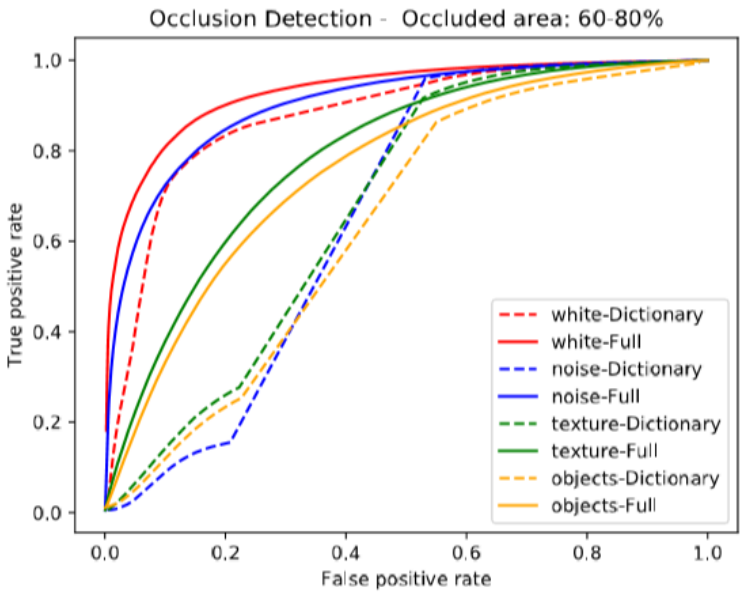


Compare CompNets to detect/localize occluders

- 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 %.







Summary

- Part 1. We started the internal representations within deep nets, by using clustering to detect Visual Concepts (VCs).
- This validated that deep nets had internal representations of object parts. We showed that the VCs could be used to detect key-points and semantic parts.
- We showed that compositional models – VC plus spatial relations – could detect semantic parts better than deep nets if there was significant occlusion.
- Part 2. We developed CompNet architectures that could classify vehicles with significant occlusion. These models were interpretable. They could detect/localize occluders, localize subparts (VCs).
- See poster in Neural Architectures Workshop (28/Oct).