Straight to Shapes: Real-time Detection of Encoded Shapes

Saumya Jetley * Michael Sapienza * Stuart Golodetz Philip H.S. Torr
Department of Engineering Science University of Oxford
{sjetley,mikesapi,smg,phst}@robots.ox.ac.uk

Abstract

Current object detection approaches predict bounding boxes, but these provide little instance-specific information beyond location, scale and aspect ratio. In this work, we propose to directly regress to objects’ shapes in addition to their bounding boxes and categories. It is crucial to find an appropriate shape representation that is compact and decodable, and in which objects can be compared for higher-order concepts such as view similarity, pose variation and occlusion. To achieve this, we use a denoising convolutional auto-encoder to establish an embedding space, and place the decoder after a fast end-to-end network trained to regress directly to the encoded shape vectors. This yields what to the best of our knowledge is the first real-time shape prediction network, running at 35 FPS on a high-end desktop. With higher-order shape reasoning well-integrated into the network pipeline, the network shows the useful practical quality of generalising to unseen categories that are similar to the ones in the training set, something that most existing approaches fail to handle.

1. Introduction

Automatically detecting [12] and delineating [6] object instances in images is a core problem in computer vision, with wide-ranging applications. For example, knowing the individual boundaries of nearby objects can allow robots to grasp them [30]. To help visually-impaired people become more independent, specially-designed glasses can highlight the boundaries of objects with which to interact [17].

For these applications, real-time frame processing is crucial, and the information provided by bounding box predictions [33] or a pixel-wise semantic segmentation of the scene [39] is not enough. A bounding box captures no more than the location, scale and aspect ratio of an object, a fairly coarse representation that provides no information about the object’s boundary. On the other hand, bottom-up pixel-labelling approaches have no explicit notion of local and global object shape. To achieve adherence to local boundaries and impose spatial and appearance consistency in the semantic space, recent works [3, 39] post-process using conditional random fields (CRFs), but this quickly becomes intractable for higher-order constructs such as global object shape, structure, pose and occlusions.

In response to the above-mentioned shortcomings, we implement an embedding space that incorporates notions of object shape, pose and occlusion patterns. A deep regression network is trained to map input image patches to this embedding space to tease apart an object’s category and its shape mask. Fig. 1 shows an example of our network regressing to three different shape representations. Observably, object shape and category are correlated for many classes, so training the learner with a single objective function that includes both terms is mutually-reinforcing.

The embedding space itself is learnt on the binary instance masks in a class-agnostic manner. By design, the embedding space is compact, decodable (supports mapping of binary instance masks in and out of the space), continuous (affords a semantically-graceful degradation of category-level instance masks around the point of interest in the space) and interpretable (one in which we can reason about the similarities between shapes and their corresponding categories). Our formulation is thus able to leverage shape reasoning and extend the prediction of shape masks to object categories our network has never seen before, but

*Authors contributed equally
which have similar characteristics to the ones seen during training, like the tiger, bear and helicopter in Fig. 2.

It is also important to note that we tackle the detection of shapes in a top-down paradigm with a single sweep of the network. This is in contrast to recent works [7, 1] combining top-down and bottom-up paradigms with various sequential arrangements of bounding-box detection, pixel-wise segmentation and category recognition to achieve instance segmentation. Due to sequential processing, the error in the above networks is additive and irreparable. For instance, if bounding-box detection comes first and cuts across a projected object boundary, pixel-wise labelling inside the box can never retrieve the correct boundary. Likewise, if detection comes after pixel-wise labelling, there is equal difficulty in delineating object boundaries between overlapping instances of the same class.

The proposed approach overcomes this inconvenient boundary clipping error by directly regressing to a higher-dimensional shape representation for each bounding box prediction, whilst using the bounding box size only for scaling. Examples demonstrating this direct regression to multiple shapes can be seen in Fig. 3. Moreover, the single-sweep processing affords our implementation real-time capabilities. To the best of our knowledge, this is the first real-time shape prediction network, running at 35 FPS on a high-end desktop with an i7-4960 Processor (3.6GHz, 12-core) and a Titan X GPU.

**Contributions.** We develop the proposed work at the intersection of object detection [32], instance segmentation [16] and joint embeddings [21], and propose a fully-differentiable network to regress directly to object locations, shapes and categories. Crucial to the process is the learning of a compact and decodable embedding space in which shapes can be described and compared. To this end, we demonstrate the use of a denoising auto-encoder [37] in encoding real-world shape templates, such as the ones found in the PASCAL VOC [10] dataset.

**2. Related Work**

**The importance of shape cues.** In the past, shape cues have been used successfully to guide object recognition [2] and localisation [19], by virtue of their ability to capture discriminative, category-specific details. With the advent of deep networks, suitable features best adapted to the task are assumed to be learnt by the network during training [20]. Learning of these features is strictly guided by the supervision goal (i.e. the optimisation loss). In existing object detection pipelines, this loss is computed on the 4D bounding-box specifications [32], thereby ignoring the use of shape to guide learning. By contrast, regressing to detailed shape vectors affords the network a more informed supervision.

**Predicting object shape masks.** The need to predict object shape masks has previously been identified as an important practical challenge in [11, 26]. However, these approaches contain independently-optimised processing blocks and have shown limited success, working only for images containing only a single object and/or with high background contrast. More recently, Pinheiro et al. [28, 29] have explored the practical benefits of learning to predict object proposals as pixel-wise segmentation masks. In [28], for example, they improve the recall scores both for bounding box and mask-based proposals. However, their networks are composed of disjoint stages for predicting segmentation masks, object locations and object classes. By contrast, and in a similar spirit to YOLO [32], the architecture that we propose for detecting shape embeddings (from which the shape masks can be reconstructed) is simpler and contains only one network, with a single objective.

**Extending bounding box detectors to shape.** Recent progress in bounding box object detection has been fuelled by the representational power of deep networks. R-CNN [13] made massive leaps in detection accuracy by using a deep network to predict the category of pre-generated bounding box proposals. Soon after, Faster R-CNN [33] emerged. It does away with separate stages for region proposal generation and classification, instead using a region proposal network that shares convolutional features with a detection network, achieving benefits in computational efficiency and accuracy. More recently, the YOLO [32] detection network streamlined object detection by replacing
the separate proposal generation and classification stages with regression to spatially-separated bounding boxes and class probabilities. This resulted in real-time object detection rates. Inspired by YOLO, we extend the network to predict encoded shapes, achieving similar speeds, but with the advantage of incorporating additional shape information.

**Shape embeddings.** Inspired by the way in which [27] learnt word embeddings to find a feature space in which similar words are close together, we aim to learn a shape embedding in which to reason about the shape of objects. In contrast to [27], however, we seek a lower-dimensional embedding (compared to the dimensions of a binary shape mask) to make detection by regression feasible.

Due to the recent successes of learned latent representations for small binary images of digits and faces [18], we learn a shape embedding by training an auto-encoder. The Shape Boltzmann Machine (SBM) of [9] uses a similar architecture to model shape space, one that is realistic and generalisable. The work of [8] extends the Shape Boltzmann framework to generate parts for prominent objects in true-colour images. However, both of these networks are trained to model shapes and are not optimised for reconstruction per se. Furthermore, it is important to note that the fully-connected construction of SBM restricts it to small synthetic images with a resolution of around $50 \times 50$. To circumvent this practical limitation, we incorporate convolutional layers, and train the model as a denoising auto-encoder [37]. Thus, in a first, we extend the full benefit of auto-encoders to real-world shape templates. By using state-of-the-art auto-encoders to learn a decodable shape embedding, we are able to predict masks for categories that have not been observed previously but share close shape similarity with those in the dataset (see Fig. 2). In addition, we are able to achieve this accuracy at a fraction of the size of hand-crafted representations.

Our work is also related to that of Li et al. [21], in which a CNN was trained with pairs of images and 3D shape embedding coordinates. However, [21] only handles single image-to-embedding mappings [36], and assumes that objects in the input image are prominent. By contrast, we regress to multiple shape embeddings, thereby extending the approach to object detection (see Fig. 3). Another important distinction is that the shape embeddings in [21] were hand-crafted and non-decodable, preventing the synthesis of new shapes from arbitrary coordinates in the embedding space and limiting the applications to retrieval with nearest-neighbour search.

**Potential applications: instance segmentation and occlusion handling.** Recent works [14, 6] on instance segmentation attempt to combine the benefits of the top-down and bottom-up paradigms by refining bounding box predictions with pixel-wise mask labelling within each box. Such bounding box-based approaches err primarily for occluded objects and their boundaries. Also noteworthy is the work by Dai et al. [5], which extends a fully-convolutional network (FCN) [24] to predict position-sensitive instance score maps that then need to be assembled into object instances at the output. Another approach [35] makes use of a recurrent neural network (RNN) architecture to predict object instances sequentially. The RNN is trained with a single loss function that is invariant to the ordering of object instances. This approach has shown promise on a limited number of object categories, but finds it difficult to remember all of the instances that it has predicted in the past, and therefore those that it needs to predict next.

Ultimately, describing an object as a set of pixels with an associated category has its limitations. In particular, shapes represented as pixel sets are hard to compare: there is no obvious way of computing meaningful distances between high-dimensional object masks. This is unfortunate, because an ability to compare shapes is extremely useful: the shape of an object is not merely an abstract property, but can indicate something fundamental about the object’s capabilities and role. For example, whilst there is significant variety amongst the many legged animals in the world, they are clearly more similar to each other than to an inanimate object such as a lorry, and the shapes involved reflect this. As such, if we can find a different representation in which we are able to compare shapes quantitatively, we can use the distance values to provide us with important clues about objects, even in categories on which we have not trained.

Additionally, bottom-up approaches are at a huge disadvantage when it comes to reasoning about object parts and their arrangements, especially when the parts are split by an occlusion. They employ post-hoc object-level or category-specific reasoning to combine the separated parts of object instances [4]. By contrast, we tackle the problem as part of an end-to-end pipeline using the intermediate embedding space, which knows about realistic category-level poses, occlusions and shape priors learnt from the training data.

### 3. Proposed Approach

**Deep regression network.** An object’s shape is one of its fundamental properties, which, when coupled with its location, scale and category, gives rich information about its coarse depth and possible pose, and how one can interact with it. To take advantage of this, we therefore extend the state-of-the-art YOLO [32] object detection network that we mentioned in §2 to regress to not only object locations, confidence scores and conditional probabilities for each category, but also detailed shape encodings, as illustrated in Fig. 4. The details of how we do this are described in §4.

**Decodable shape representation.** We require a shape embedding that is both compact and decodable for the object shape prediction application. Moreover, we would benefit immensely from a representation that embodies higher-
order understanding of object shape, realistic poses and occlusion patterns. For instance, we would like the distances between shapes in the representation to reflect our own understanding of the similarity between shapes. To this end, we investigate three decodable shape representations: i) downsampled binary masks, ii) radial vectors and iii) learned shape encodings (see §5).

In §6, we join the ideas from §4 and §5, and evaluate our real-time shape detection pipeline on the data provided in the PASCAL VOC and SBD [10, 15] datasets.

4. Deep Regression Network

This section draws inspiration from the YOLO detection algorithm [32] and extends it for the prediction of shapes. Like YOLO, our network reasons globally about objects in the image and is trained to minimise a single objective function. However, we augment YOLO’s object representation with an encoded vector that denotes an object’s shape.

The YOLO pipeline starts by dividing the input image into an $S \times S$ grid. If the centre of an object lies within a cell, then that cell becomes responsible for detecting that object. Here, each grid cell predicts $B = 2$ shapes, boxes and confidence scores, as well as a probability mass function. The confidence scores are calculated as the intersection over union (IoU) between the predicted and target boxes, if a target box exists, and zero otherwise. Each shape encoding has $N$ parameters and represents the shape of an object independently of its location and aspect ratio.

For instance, in order to calculate the binary shape mask we provide as a target for learning, we first extract a minimum bounding box around the ground truth object segmentation; this captures the location and aspect ratio of the object. Then, the object segmentation is binarised, cropped, and rescaled as part of the encoding process.

Each grid cell also predicts a conditional probability mass function, which when multiplied by the object confidence score, results in a score reflecting the confidence for the category and its overlap:

$$p(c|o) * p(o) * \text{IoU} = p(c) * \text{IoU}. \quad (1)$$

As input to our network, we feed in $448 \times 448$ images, which are transformed using random rotations, translations, spatial scaling, and pixel scaling. The network’s target is constructed by transforming the ground truth shapes with the same geometric augmentation transformations. The final size of the target vector for a single image becomes:

$$D = S \times S \times (N \times B + |C|), \quad (2)$$

where $|C|$ is the number of dataset categories, and $N$ includes parameters for the shape score, minimum bounding box, and encoding. For instance, if the shape is to be encoded by a $16 \times 16$ binary mask, then $N = (1 + 4 + 256)$.

For training and inference, we use Darknet [31]. We optimise the sum of squared errors between the predicted network output and a target tensor. The loss function used to train the network has the following components:

$$L = L_{\text{box}} + L_{\text{conf}} + L_{\text{shape}} + L_{\text{pmf}}, \quad (3)$$

where

$$L_{\text{shape}} = \lambda_{\text{shape}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \sum_{k=0}^{N-5} \mathbb{1}_{ij}^{|*|}(\tau_{ijk} - \tilde{\tau}_{ijk})^2. \quad (4)$$

Definitions for the other loss terms can be found in [32]. Since a prediction only requires one forward pass of the network, it is very fast at test time (§6). We use non-maximal suppression to prune the 96 shapes output by the network, which helps to reduce multiple overlapping detections.

5. Decodable Shape Representation

The shape embedding space is a crucial part of the proposed formulation. We agree that separating the tasks of
constructing the embedding space and learning a mapping from the image domain to this embedding makes the task more tractable [21]. Moreover, we believe that they are intrinsically independent tasks and should therefore be treated separately. We thus experiment with two hand-crafted and one learned shape representation to establish the embedding space, described as follows.

5.1. Downsampling binary shape masks

A very simple shape descriptor can be obtained just by downsampling the full-size binary shape mask. Given a fixed descriptor size $d = k \times k$, we resize the image down using OpenCV’s pixel area relation-based resampling approach. For reconstruction, the descriptor can be resized back up again using bicubic interpolation. For large values of $d$ such as 256, this descriptor can achieve respectable IoU scores for reconstruction, but for small values of $d$ such as 25, it struggles to adequately represent non-trivial shapes.

5.2. Radial representation

A radial descriptor represents a shape as a series of distances between some centre point within the shape and points deterministically distributed over the shape’s boundary. These denote how far the shape extends in particular directions from the centre point. Since the descriptor needs to provide information about the shape’s extent in every direction, it is common to choose the boundary points by finding where rays cast outwards from the centre point at angles uniformly distributed over $[0, 2\pi)$ intersect the boundary (alternatively, rays may be cast inwards towards the centre point). The choice of centre point is arbitrary, but to improve shape reconstruction it makes sense to choose a centre point that has a direct line of sight to as much of the boundary as possible. In practice, to achieve better IoU scores for reconstruction, we construct several radial descriptors for each shape and pick one with maximal IoU. More details can be found in the supplementary material.

5.3. Learned shape encoding

The aim is to implement a network that can progressively compress an input binary mask to a comparatively lower dimensional space. The autoencoder has already been shown to improve over PCA and logistic PCA at a similar task for digits, curves and faces [18].

Thus, in an initial experiment, we use the fully-connected autoencoder architecture (784-1000-500-250-30) of [18] to learn an embedding space for Caltech-101 silhouettes. We train the model for a shape space using the silhouettes contained in the trainval set. In addition to the RBM style pre-training of [18], we fine-tune the network weights using cross-entropy error between the predicted and the target shape masks. This setup yields a IoU test reconstruction accuracy of 0.85. It becomes evident that this architecture, also employed in in [9, 8], is not able to exploit the spatial redundancy in visual data due to its fully-connected nature. For the same reason, it is not scalable to large images.

It is important to note that similar shape patterns or boundary formations can be found at different locations in the input image, for example the front and rear tyre of a bicycle or bike look similar. We can leverage this repetition of visual patterns by the use of a convolutional construct. Accordingly, a block diagram of our autoencoder network is shown in Fig. 5. To prevent the network from learning an identity mapping to a higher-dimensional space, we introduce white noise in the input image so the network generalises to a denoiser [37]. We minimize the cross-entropy loss to adjust the network weights.

6. Experiments and Results

Our method introduces a new perspective to the challenge of scene understanding, whereby higher-order concepts of shape are integrated into the end-to-end prediction pipeline. This allows us to unify detection and segmentation. We thus study the application of our proposed approach to instance segmentation (§6.2). We find that training the object detection network with the concept of shapes allows it to predict shape masks for unseen categories at test time. We compare our qualitative results to those produced by a state-of-the-art instance segmentation method (§6.3).

Essential to our formulation is the need to reconstruct accurate shape masks from a low-dimensional shape representation. We thus analyse the reconstruction quality of three different shape spaces in the following section (§6.1).
6.1. Comparing the Shape Representations

The choice of representation is crucial, as it dictates what aspects of the shape are made explicit and the ease with which it can be manipulated [25]. We thus compare the shape representations outlined in §5 on three different axes: i) their ability to accurately represent shapes at low dimensionalities, ii) the continuity of the associated shape space and iii) the structure of the shape space.

Representation ability: We compute the average shape reconstruction error (one minus the IoU) yielded by each of our three descriptors at different sizes on the SBD dataset. Table 1 shows how the error for each descriptor varies with the representation size. In line with intuition, at low descriptor dimensionalities, our encodings are better able to preserve shape than either of the two hand-crafted alternatives. However, the benefits fade away at larger dimensionalities. A qualitative example of the artifacts introduced by lowering the dimensionality of each shape representation can be seen in Fig. 6: both of the two hand-crafted representations experience a severe decrease in quality as the size decreases, whereas our representation still manages to capture the major topological features of the shape even at size 20.

Shape space continuity: To study the continuity of shape spaces, we add Gaussian noise to the encodings and see its effect on the reconstructed shape masks in Fig. 7. As expected, an increase in noise degrades all representations; however, a little bit of noise in the radial vector distorts the shape beyond recognition. The binary mask loses precision at the boundaries and the wheels of the bicycle blend into the border of the image. The learned shape representation is able to preserve details such as the seat, and the circular wheels better than the hand-crafted alternatives.

Shape space structure: We visualise the structure of the shape spaces associated with two representations – the learnt encodings and the radial descriptors – in Fig. 8. Even at a first glance, it is obvious that the shape space for the learnt encodings provides a better category-based clustering of the shape masks. For instance, as we move from the top left region of the space to the bottom right along a diagonal we observe that we transition from samples with large boundary projections (horses and birds) to upright shapes of humans and finally to more streamlined shapes such as bottles. There is no such clear structure in the shape space for circular descriptions: indeed, the categories seem to be scattered across various disparate clusters. We also find and visualise the nearest neighbour of each of a pre-selected set of 10 ‘anchor’ images. Most interesting to note are the ground truth masks of the neighbouring representations of anchors 2, 4, 7 and 10. The learnt embedding space observably shows a closer match between neighbouring shapes.

![Figure 6](image1.png)

Figure 6: The effect of reducing the dimensionality of each shape representation. From left, the original image, binary masks, radial vectors, and a learned embedding. As the dimensionality is reduced, both the binary masks and radial vectors lose precious details such as the legs and head of the bird, whilst the learned embedding is able to preserve the overall shape and degrades more gracefully.

![Figure 7](image2.png)

Figure 7: The effect of adding random noise at various levels to each of the shape representations.

| SBD          | 20 (25) | 50 (49) | 100 | 200 (196) | 256 |
|--------------|---------|---------|-----|-----------|-----|
| Downsampling mask | 0.15    | 0.10    | 0.07| **0.05**  | **0.04** |
| Radial       | 0.13    | 0.08    | 0.07| 0.06      | 0.06 |
| Shape encoding | **0.10**| **0.07**| **0.06**| **0.05** | **0.05** |

Table 1: The average shape reconstruction error (one minus the IoU) achieved by each of our three descriptors at various different sizes on the validation portions of the SBD [10, 15] dataset. The unbracketed numbers in each column indicate the sizes used for the radial and shape encoding descriptors; the bracketed numbers indicate the sizes used for the downsampled mask descriptors, which are square by design. The lowest error(s) in each size category are highlighted in bold.

6.2. Instance Segmentation as an Application

The hypothesis that we would like to test is that we are able to regress directly to an object-based representation of the scene. For this experiment, we train a YOLO-style network that maps directly from a single image to a collection of objects. Since we have the locations and shape encodings of the objects, we are able to decode their shapes and generate labelled segmentation masks that can be compared...
Figure 8: 2D visualisations of the 20D learned embedding space (left) next to the 256D radial vector space (right) with t-SNE. Notice the difference in the anchor’s nearest neighbours between both representations. For instance no.10 class bicycle is facing left in the anchor, the nearest neighbour in the learned embedding is also facing left even though it is occluded, whilst the neighbour of no.10 in the radial space is facing to the right.

against those generated by state-of-the-art instance segmentation methods [16, 1, 5].

**Dataset, splits and performance metrics:** We train and evaluate our method on the established PASCAL VOC with SBD annotations [10, 15] dataset, with the same split sets use by [16, 5]. We report the mean average precision at various overlap thresholds (AP, APR$_{vol}$) [16].

**Results and discussion:** In Table 2, we report the instance segmentation results we obtain when regressing to the following shape representations: i) downsampled binary shape masks ($d = 256$), ii) radial vectors ($d = 256$), and iii) learned embeddings ($d \in \{20, 50\}$). The dimensionalities were chosen based on the reconstruction errors obtained in Table 1, where we observed that we are able to use lower-dimensional shape codes with the learned embedding as compared to the hand-crafted ones (see Fig. 6). This reduction in the number of parameters per shape means that we can train neural network models with fewer parameters, and with less chance of overfitting. This intuition is consistent with our empirical findings in Table 2, where we achieve better performance as the dimensionality of the embedding is reduced from 50 to 20. In comparison to both of our embeddings, both downsampled $16 \times 16$ binary masks and $256D$ radial descriptors perform worse, despite having more space available in which to represent the shape. One possible reason for this in the case of radial descriptors is their higher sensitivity to noise, as can be seen in Fig. 7. Qualitative results showing the performance of our method on the SBD dataset can be seen in Fig. 10.

It will be seen in Table 2 that our instance segmentation results lag somewhat behind current state-of-the-art methods; this parallels the lag in object detection results experienced by YOLO with respect to state-of-the-art methods in that field. We hypothesise that this is because both approaches struggle to precisely localise certain objects, especially small ones (see Figure 10). However, like YOLO, our approach runs in real time, something that cannot be achieved by any of the offline methods against which we compare (as can be seen in Table 2, existing methods are in no way real time, taking at least an order of magnitude longer to run). We are also much better than the state-of-the-art instance segmentation method of Arnab et al. [1] at recognising unseen categories (see Fig. 11), as discussed in the next section.

### 6.3. Zero-shot segmentation

Traditionally, the task of zero shot learning has been achieved through the use of attributes [34, 38]. A recent work on zero shot boundary segmentation [22] takes a similar tack. By constrast, the proposed approach doesn’t need
Figure 10: (a) Correct predictions using the downsampled binary mask descriptors. (b) Correct predictions using 20-dimensional learnt shape encodings. In the 3rd image the horns of the cow are missed while the human shape mask gets elongated due to an incorrect bounding box prediction. (c) Examples of missed detections using the 20-dimensional shape encodings. The network misses out or false fires on small objects (images in the 2nd column. The dogs in the images are falsely categorized as cats, while the sofa incorrectly includes the nearby dining table.

Table 2: Quantitative instance segmentation results on PASCAL SBD 2012 val. The timing results were obtained on a high-end desktop containing a Titan X.

| Method            | AP r@.5 | AP r@.7 | AP r@.75 | Time (ms) |
|-------------------|---------|---------|-----------|-----------|
| BinaryMask        | 32.3    | 12.0    | 28.6      | 26.3      |
| Radial            | 30.0    | 6.5     | 29.0      | 27.1      |
| Embedding (50)    | 32.6    | 14.8    | 28.9      | 30.5      |
| Embedding (20)    | 34.6    | 15.0    | 31.5      | 28.0      |
| SDS [16]          | 49.7    | -       | 41.4      | 48k       |
| MNC [3]           | 65.0    | 46.7    | -         | 330       |

Table 11: A comparison between our shape detection results (c) and the state-of-the-art semantic segmentation (b) and instance segmentation (e) algorithm by [1] on images of animals (a) taken from YouTube videos which are not present in the PASCAL VOC training set. In the first two rows, the instance segmentation predicts that the legs of the tiger are human. Also note that our shape prediction method is more consistent over the tiger images taken from the same video. In the lower rows, the instance segmentation (c) fails to predict any segments, whilst our method predicts class ‘dog’ for the tiger, hedgehog, baby elephant and bear, and class ‘horse’ for the large elephant.

tiger to a cat or dog-like animal. This is exactly how our network approaches this task.

7. Conclusion

In this work, we show for the first time that it is possible to regress directly and simultaneously to multiple object representations that incorporate the notion of shape. We combine ideas from object detection, instance segmentation and low-dimensional embedding spaces to create a real-time system able to detect encoded shapes. One key factor that allows us to do this is the introduction of a shape embedding space that is both compact, decodable, continuous and interpretable. We find that imbuing object detectors with a knowledge of shape allows us to predict plausible shapes for previously-unseen dataset categories, allowing us to detect the presence of objects in the image for which current state-of-the-art instance segmentation methods perform poorly or fail.

Our next step is to investigate how our shape prediction approach can be extended to cope with a larger variety of shapes and object categories like those in MS-COCO [23].
References

[1] A. Arnab and P. H. S. Torr. Bottom-up instance segmentation using deep higher-order crfs. In Proc. British Machine Vision Conference, 2016. 2, 7, 8
[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell., 24(4):509–522, Apr. 2002. 2
[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv preprint arXiv:1606.00915, 2016. 1
[4] Y.-T. Chen, X. Liu, and M.-H. Yang. Multi-instance object segmentation with occlusion handling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3470–3478, 2015. 3
[5] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. CoRR, abs/1603.08678, 2016. 3, 7, 8
[6] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. CoRR, abs/1512.04412, 2015. 1, 3
[7] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. arXiv preprint arXiv:1512.04412, 2015. 2
[8] S. Eslami and C. Williams. A generative model for parts-based object segmentation. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 100–107. Curran Associates, Inc., 2012. 3, 5
[9] S. A. Eslami, N. Heess, C. K. Williams, and J. Winn. The shape boltzmann machine: a strong model of object shape. International Journal of Computer Vision, 107(2):155–176, 2014. 3, 5
[10] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. Pascal visual object classes challenge results. Available from www.pascal-network.org, 2005. 1, 2, 4, 6, 7
[11] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. Int. Journal of Computer Vision, 2009. 2
[12] R. Girshick. Fast R-CNN. In Proc. Int. Conf. Computer Vision, 2015. 1
[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2014. 2
[14] S. Gupta, R. B. Girshick, P. A. Arbeláez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In Proc. European Conf. Computer Vision, 2014. 3
[15] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In Proc. Int. Conf. Computer Vision, 2011. 4, 6, 7
[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In Proc. European Conf. Computer Vision, 2014. 2, 7, 8
[17] S. L. Hicks, I. Wilson, L. Muhammed, J. Worsfold, S. M. Downes, and C. Kennard. A depth-based head-mounted visual display to aid navigation in partially sighted individuals. PLoS ONE, 8(7):1–8, 07 2013. 1
[18] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, 2006. 3, 5
[19] A. K. Jain, Y. Zhong, and S. Lakshmanan. Object matching using deformable templates. IEEE Transactions on pattern analysis and machine intelligence, 18(3):267–278, 1996. 2
[20] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10):1995, 1995. 2
[21] Y. Li, H. Su, C. R. Qi, N. Fish, D. Cohen-Or, and L. J. Guibas. Joint embeddings of shapes and images via CNN image purification. In Siggraph Asia, 2015. 2, 3, 5
[22] Z. Li, E. Gavves, T. E. J. Mensink, and C. G. M. Snoek. Attributes make sense on segmented objects. In European Conference on Computer Vision, 2014. 7
[23] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In Proc. European Conf. Computer Vision, 2014. 8
[24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional models for semantic segmentation. In IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2015. 3
[25] D. Marr. Vision. W. H. Freeman and Company, 1982. 6
[26] M. Marszalek and C. Schmid. Accurate object localization with shape masks. In IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2007. 2
[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013. 3
[28] P. O. Pinheiro, R. Collobert, and P. Dollár. Learning to segment object candidates. In Advances in Neural Information Processing Systems, 2015. 2
[29] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In Proc. European Conf. Computer Vision, 2016. 2
[30] D. Rao, Q. V. Le, T. Phoka, M. Quigley, A. Sudsang, and A. Y. Ng. Grasping Novel Objects with Depth Segmentation. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pages 2578–2585, 2010. 1
[31] J. Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013–2016. 4
[32] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In IEEE Int. Conf. on Computer Vision and Pattern Recognition, 2016. 2, 3, 4
[33] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems, 2015. 1, 2
[34] B. Romera-Paredes. An embarrassingly simple approach to zero-shot learning. In Int. Conf. on Machine Learning, 2015. 7
[35] B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. CoRR, abs/1511.08250, 2015. 3
[36] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. Structured prediction of 3d human pose with deep neural networks. In Proc. British Machine Vision Conference, 2016. 3
[37] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning, pages 1096–1103. ACM, 2008. 2, 3, 5
[38] Z. Zhang and V. Saligrama. Zero-shot learning via joint latent similarity embedding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6034–6042, 2016. 7
[39] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional Random Fields as Recurrent Neural Networks. In Proc. Int. Conf. Computer Vision, pages 4321–4329, 2015. 1