Visual Semantic Context Encoding for Aerial Data Introspection and Domain Prediction

Andreas Kriegler[1,2][0000−0002−5653−5181], Daniel Steininger[1][0000−0003−3810−1803], and Wilfried Wöber[3,4][0000−0002−0881−205X]

[1] Vision Automation and Control, Austrian Institute of Technology, 1210 Vienna, Austria
[2] Visual Computing and Human-Centered Technology, TU Wien Informatics, 1040 Vienna, Austria
[3] Industrial Engineering, UAS Technikum Wien, 1200 Vienna, Austria
[4] Integrative Nature Conservation Research, University of Natural Resources and Life Sciences, 1180 Vienna, Austria
andreas.kriegler@ait.ac.at

Abstract. Visual semantic context describes the relationship between objects and their environment in images. Analyzing this context yields important cues for more holistic scene understanding. While visual semantic context is often learned implicitly, this work proposes a simple algorithm to obtain explicit priors and utilizes them in two ways: Firstly, irrelevant images are filtered during data aggregation, a key step to improving domain coverage especially for public datasets. Secondly, context is used to predict the domains of objects of interest. The framework is applied to the context around airplanes from *ADE20K-SceneParsing*, *COCO-Stuff* and *PASCAL-Context*. As intermediate results, the context statistics were obtained to guide design and mapping choices for the merged dataset *SemanticAircraft* and image patches were manually annotated in a one-hot manner across four aerial domains. Three different methods predict domains of airplanes: An original threshold-algorithm and unsupervised clustering models use context priors, a supervised CNN works on input images with domain labels. All three models were able to achieve acceptable prediction results, with the CNN obtaining accuracies of 69% to 85%. Additionally, context statistics and applied clustering models provide data introspection enabling a deeper understanding of the visual content.

Keywords: context encoding · domain prediction · aerial scenes

1 Introduction

Humans intuitively incorporate contextual information when trying to understand the environment they perceive. Objects appearing in an unfamiliar semantic context or out-of-context objects [7], such as airplanes on a highway, attract the observer's attention since they are typically related to other scenes.

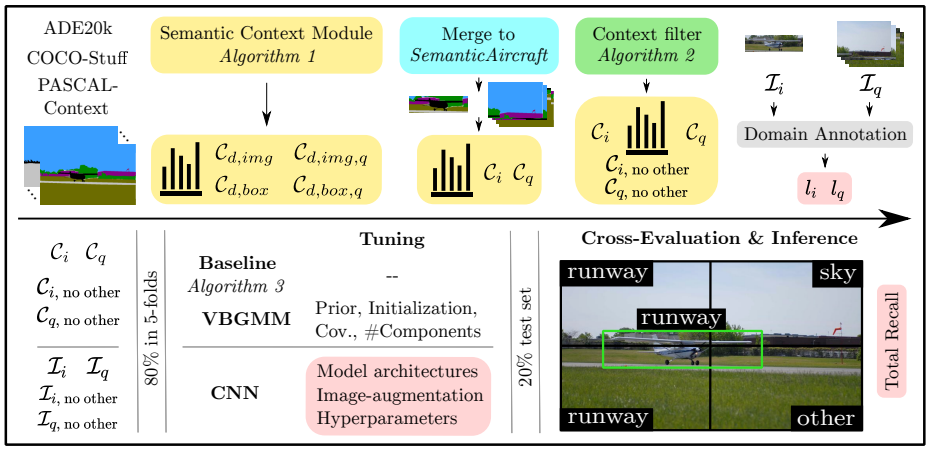

Fig. 1: Upper half: For every source dataset d, context vectors $\mathcal{C}$ on images (img) and object bounding boxes (box) as well as the respective quadrants q are obtained. Merging leads to $\mathcal{C}_i$ and $\mathcal{C}_q$ for instances and quadrants, removal of other-patches yields $\mathcal{C}_{i,no\_other}$ and $\mathcal{C}_{q,no\_other}$. All image patches $\mathcal{I}_i$ and $\mathcal{I}_q$ were then annotated with domain labels l_i and l_q. Lower half: Either the set of context vectors or images with annotations were finally used for domain prediction.

Incorporating this kind of prior information has the potential to improve computer vision (CV) models by assigning meaning to objects and actions, enabling "visual common-sense" and is essential for solving upcoming challenges in scene understanding [1]. In particular, autonomous systems operating in the real world struggle to stay robust when traversing multiple environments, or the surroundings look significantly different due to weather, atmospheric effects, or time of day. Semantics of images or semantic parsing in the field of CV refers to the recognition and understanding of the relationship between objects of interest other objects and their environment [7]. Natural occurrences of objects and corresponding environments are analyzed to transfer this information into a logical-form representation, understandable for machine vision systems. This context can be understood as a statistical property of our world [19]. On a micro level semantic segmentation yields information regarding both foreground objects, commonly referred to as *things* [17] and background scenery, known as *stuff* [4]. Following this segmentation and applying ideas from natural language processing semantic relations between *things* and *stuff* can be formulated. It is well known that cues stem from the *semantic context* surrounding objects and this visual context is therefore a necessity for more complete scene understanding [5]. The models developed in this work are evaluated in the field of avionics, specifically on images showing airplanes. Therefore, the related concept of *domains* in this work holds two specific yet congruent meanings: In the applied sense, a domain describes the local real world surroundings of airplanes, in the more formal sense a domain can be understood as a collection of characteristic classes. The syn-

tactic evolution from semantic context to domains is natural, when considering the focus lies on a person, object or autonomous agent around which context is formulated. Domains can therefore be understood as a result of the analysis of an objects semantic context, placing *things* into a distinctive domain. Following the analysis of this visual semantic context, characteristic statistics can give an understanding of datasets which can in turn be used to guide data-aggregation strategies. Further along the learning pipeline, due to the convolutional kernel in convolutional neural network (CNN), contextual information is usually learned implicitly regardless of the learning task at hand [2,26]. Contextual information is embedded in the feature space and the learned kernel-parameters lead to the well-known bias/variance dilemma [12]. An explicit representation of context in the form of distinct domains might allow intelligent systems to swap between model parameters in a mixture-of-experts fashion. To this end the preliminary step of predicting domains is studied (see Figure 1 for an overview of the proposed methods). To summarize, this work makes the following contributions:

1. We propose a simple method to encode semantic context from segmentation masks providing context insights for images from the aerial domain.
2. We present the merged dataset *SemanticAircraft* and filter images using context statistics. Additionally, we provide over 17k domain annotations for *SemanticAircraft*.
3. We use unsupervised clustering algorithms for data introspection revealing further information relevant to avionic applications. Finally, we reinterpret clustering results for domain prediction, propose a novel, fast and interpretable prediction algorithm as baseline and compare these results to domain classification results using deep supervised CNNs.

The remainder of the paper is structured as follows: Section 2 provides related works. Section 3 details the encoding algorithm, shows context results and explains the domain annotation process for *SemanticAircraft*. Section 4 introduces the three domain prediction models and provides inference results.

2 Related Works

Following classification, object detection and semantic segmentation, a clear trend towards more complex representations is noticeable [16,32]. Before CNNs, semantic segmentation used either conditional random fields (CRFs) or tree models similar to Markov networks [22,23], although with limited accuracy.

2.1 Semantic Segmentation with Deep Learning

In a similar vein to works using CRFs is Wang *et al.* [29]'s multiple-label classification on *NUS-WIDE* [8], *COCO* [17] and *VOC* [10]. They combine a VGG [28] CNN to embed visual features with a LSTM-RNN for label information in a joint space. Zhang *et al.* [34] pose the question whether capturing contextual information with a CNN is the same as simply increasing the receptive field size but

perhaps more accurately the question should be how much one can increase the receptive field size and still capture relevant contextual information. In a similar vein Fu *et al.* [11] state that the method with which to effectively capture pixel or region-aware context is still an open and unresolved research question. While these works provide models for obtaining semantically segmented images, neither the learned features nor final masks constitute an explicit context representation that is in line with our concept of visual domains. The literature on domain adaptation techniques [31,30,9] on the other hand is predominately concerned with domain adaptation between synthetic and real images for transfer learning. For the purpose of making a segmentation network robust across multiple domains, Chen *et al.* [6] propose to treat different cities as distinct domains and go on to learn both class-wise and global domain adaptation in an unsupervised manner. The concept of domains are treated as a means-to-the-end for boosting segmentation accuracy which is a common approach. Similarly, Sakaridis *et al.* [24] use the idea of guided curriculum model adaptation for improving semantic segmentation of nighttime images for advanced driver assistance systems (ADASs). Having captured the same scene at daytime, twilight and night using labeled synthetic stylized and unlabeled real data, models are transferred from daytime to night with twilight as an intermediate domain, using the Dark Zürich dataset. In a similar vein are the works of Zhang *et al.* [36] and their follow-up paper [35]. They learn global label distributions over images and local distributions over landmark superpixels and feed those into a segmentation network to boost semantic segmentation performance.

2.2 Domains for Context Generalization

While these works provide a foundation for capturing context in a deep learning (DL) manner, and also tackle the problem of domain generalization, it stands to reason that the semantically segmented output masks are much lower-level in their representation of the context than desirable. It could be argued that pixel-wise classification as final model output is less representative of the actual content of an image than our conceptual usage of domains around target objects, at least for object-centric tasks. To this end the work of Sikirić *et al.* [27] is related. The task is image-wide classification of images captured in various traffic scenes in Croatia. Their treatment of different traffic scenes is similar to the idea of domains in this work: As a concept to describe the environment for scene parsing. Although the methods developed in our work are kept as general as possible to allow the application in multiple domains, we will focus on one domain in particular, the aerial domain. While methods for ADAS applications have gotten strong interest, the aerial domain is much less studied in contemporary literature.

2.3 Public Aerial Datasets

Publicly available datasets providing semantically segmented images are fairly numerous, around 10-15 according to [14]. Nevertheless, no semantically-segmented

dataset specifically created for airplanes exists. Three semantic public datasets that feature some images of airplanes are accessible: A derivative of the *ADE20K* dataset for scene parsing [37] referred to as *ADE20K-SceneParsing*. An extension to the *MS COCO* [17] annotation for *stuff* classes [4] denoted as *COCO-Stuff*. And the semantic extension to *PASCAL-VOC* [10], *PASCAL-Context* [19]. For brevity these special derivations will be referred to as *ADE*, *COCO* and *PASCAL*. These three datasets from the basis for our following context analysis.

3 Semantic Context and *SemanticAircraft*

In this section we outline our taxonomy of aerial domains, detail the data aggregation process, introduce our algorithm to compute context vectors and use context statistics and a context filter to merge images to *SemanticAircraft*, for which we finally provide domain label annotations.

3.1 Aerial Domains and Aggregation of Airplane Images

Considering the environment airplanes traverse, three domains can be identified:

Apron: In aviation the area where airplanes are usually parked, loaded or unloaded with goods, boarded or refueled is referred to as apron. A large variety of partially occluded objects, persons and unusual vehicles such as mobile loading ramps, taxiing vehicles and moving stairways, is common.

Runway: The strip of asphalt or concrete used primarily for takeoff and landing of the airplanes is referred to as runway. It is usually enclosed by grass or other types of soil, with more vegetation such as bushes and trees appearing to the sides. Neither vehicles nor persons are usually encountered in this domain.

Sky: Sky is typically a smooth blue or grey background to the airplane, but clouds and time of day can significantly alter its appearance. The elevation angle of the capturing camera plays an important role.

Other: Finally we use a fourth domain, other, for out-of-context airplanes.

Images from *ADE*, *COCO* and *PASCAL* featuring at least one airplane pixel in their semantic masks were aggregated and the following observations made:

ADE20K-SceneParsing: *ADE* features 146 images with airplanes in total, where 33 of the 150 classes are of interest. Images on average are around 600×600 in size. Two pairs of duplicate images exist where only one image is kept.

COCO-Stuff: *COCO* is the largest of the three datasets with 3079 images featuring airplanes. *COCO-Stuff* has 171 classes with 41 being relevant. Average image size is around 640×480. Besides out-of-context airplanes, *COCO* also

Algorithm 1 Obtaining semantic context

Requisites: A set $\mathcal{M}$ of masks $\mathbf{m}$ mapping from pixels to the list of class ids $\mathbf{x} = \{0, \ldots, N\}$, in particular t for the target class and label v for void pixels.

function GETCONTEXT($\mathcal{M}, \mathbf{x}$)
  for $\mathbf{m} \in \mathcal{M}$ **do** ▷ For every segmantic mask
    $\mathbf{m_{ext}} \leftarrow$ dilate($\mathbf{m}$, t, 1, 5) ▷ Dilate the mask to deal with void pixels
    for x in $\mathbf{x}$ **do** ▷ For every class in consideration
      if $x \neq t, v$ **then** ▷ Ignore specific classes
        $\mathbf{c}_{\mathbf{m},x} \leftarrow \sum_{i=0,j=0}^{i=W,j=H} (\mathbf{m_{ext}}_{i,j} == x)$ ▷ Count class-specific pixels
  $\forall x \in \mathbf{x} : \mathbf{c} \leftarrow 100 \times \frac{\mathbf{c}_{\mathbf{m},x}}{\sum(\mathbf{c_m})}$ ▷ Normalize to obtain a $(0, 1]$ squashed vector
  return c

features synthetic images.

PASCAL-Context: The total number of images with airplanes is 597. It has 456 classes in total of which around 30 are applicable. Here, many variations for building and soil exist. The average image size is around 470×386.

Every image featuring at least one airplane pixel is considered. If the dataset provides bounding box (BBox) annotations they are used for extraction of instances, otherwise an algorithm iteratively extends rectangles encompassing 1 pixel at the start to include all pixels of the same target class touching any already included pixels.

3.2 Encoding Visual Semantic Context

The method used for obtaining semantic context is similar to the concept of label occurrence frequency presented by Zhang *et al.* [36]. In this work we extend the algorithm to handle semantic uncertainty boundaries, exclude undesired classes and apply the method to a finer granularity of image regions. The semantic context module described in algorithm 1 extracts label frequency for a set $\mathcal{M}$ of masks $\mathbf{m}$. Besides the masks, the list of classes $\mathbf{x} \in \mathbb{R}^{u \times 1}$ is required. As a first step, to deal with void pixels at label transitions, the boundaries of the target instance are expanded by five pixels in every direction, i.e. the instance gets dilated by 1 for five times. Then for every class in $\mathbf{x}$ the number of pixels in a certain patch of $\mathbf{m}$ is obtained and normalized with the total number of pixels in that patch. For void and airplane pixels, they can optionally be ignored. As a result, a number of context distribution vectors $\mathbf{c}$ are obtained in every dataset d. Vector $\mathcal{C}_{d,img,q=II}$ then for example gives the context in dataset d across the second image quadrants. Quadrant-I is the top-right image quadrant counting counter-clockwise. Entries in $\mathbf{c}$ are also sorted by magnitude. We deal with instances and image quadrants separately, since a downstream tracking framework would benefit from this granularity. The module was applied to the datasets *ADE*, *COCO* and *PASCAL* to obtain first statistical context measures.

Algorithm 2 Semantic context filter

Requisites: Set $\mathcal{C}$ of vectors $\mathbf{c}$ holding statistics for every patch to be filtered. Labels $\mathbf{x}$ that shall get filtered with quantile percentages $\mathbf{p} = (0.93, 0.93)$.

function FILTERCONTEXT($\mathcal{C}, \mathbf{x}, \mathbf{p}$)
  for (x, p) in $(\mathbf{x}, \mathbf{p})$ **do** ▷ For every filter label
    $q_{i,x} \leftarrow \text{quantile}(\mathcal{C}_{(i,x)}, p)$ ▷ Calculate quantiles in instances
    $q_{q=I\ldots IV,x} \leftarrow \text{quantile}(\mathcal{C}_{(q=I\ldots IV,x)}, p)$ ▷ And image quadrants
    $\mathcal{S}_{i,x} \leftarrow \mathcal{C}_{i,x} < q_{i,x}$ ▷ Filter instances using the threshold
    for $i \leftarrow I$ to IV **do**
      if $i \leq II$ **then**
        $\mathcal{S}_{q=i,x} \leftarrow \mathcal{C}_{i,x} < mean(q_{(q=I,x)}, q_{(q=II,x)})$ ▷ Or mean for quadrants
      else
        $\mathcal{S}_{q=i,x} \leftarrow \mathcal{C}_{i,x} < mean(q_{(q=III,x)}, q_{(q=IV,x)})$
  return $\mathcal{S} \setminus (\mathcal{S}_{\text{indoor}} \cup \mathcal{S}_{\text{void}})$ ▷ Let both constraints apply for the final set

3.3 Aggregation and Context of *SemanticAircraft*

When training a framework on public datasets a single pass of data aggregation does not yield optimal data in terms of domain coverage, redundancy, object size and especially class consistency. Therefore, following inspection of context statistics, semantically similar classes were merged to superclasses for *SemanticAircraft*. Bounding boxes were increased by thirty percent, which strikes a good balance of incorporating distant elements while leaving out largely irrelevant features. First context statistics have shown that many images and instances feature undesirable context traits, e.g. a high-percentage of void and indoor pixels. Since semantic context yields a high-level understanding of the scene, it can be used to filter patches where the context in specific classes is higher than a desired value (algorithm 2). To be precise, using the set $\mathcal{C}$ of context vectors for any patch, obtain the quantile value q_p at the threshold p and remove all patches with context above q_p, while using the mean for quadrants-I/II and III/IV. Our motivation for using the mean of the upper vs. lower image is, that the majority of images showing airplanes feature a clear horizon (top) or ground (bottom) separation, if the airplane is on the ground. While visual content in the lower left quadrant might be dissimilar to content of the lower right for a limited number of images, assuming a dataset of infinite size, the content of the two quadrants becomes equal. The same reasoning holds for the upper two quadrants. Therefore, to reduce dataset bias we take the mean of the upper and lower quadrants. Following the heuristic of filtering patches showing at least a small amount of indoor pixels, the percentages were determined empirically as $\mathbf{p} = (0.93, 0.93)$. This removes indoor images while also removing samples with excessive void pixels. As a last filtering step, low-level filters were applied: All instances with width or height shorter than 60 pixels and aspect ratio larger than 6:1 were discarded. This leaves 3854 instances and 13265 image-quadrants. Although not visualized in this paper, the filtered out-of-distribution patches include toy-airplanes, airplanes in

Fig. 2: A random selection of instances (top) and quadrants (bottom) from *SemanticAircraft*. Clutterness in quadrants is lower than instances, across all domains. Although looking like a sky image, the fourth image in the other row of the upper half actually shows a computer generated image. Images with red outline were misclassified by *any* of the three prediction models (see section 4).

magazines, LEGO-airplanes and many airplanes in museums and exhibitions. Without the exclusion, these samples would bring unwanted noise into the data for training domain prediction models. Figure 2 provides example images from the resulting dataset *SemanticAircraft*. Final semantic context statistics were obtained and can be seen in table 1.

4 Domain Prediction on *SemanticAircraft*

In this section we propose the application of semantic context for the task of domain prediction. Three distinct approaches were chosen for this task:

Baseline: First, define domains as set of superclasses. Using the context vectors $\mathcal{C}_i$ and $\mathcal{C}_q$ run a threshold algorithm with defined ranges and weights.

Unsupervised: Using the set of context features $\mathcal{C}_i$ and $\mathcal{C}_q$, fit an unsupervised machine learning (ML) model to predict the domain for unseen context vectors, i.e. interpret label statistics per patch as features and use unsupervised learning for clustering – reinterpret the clusters for a classification setting.

Table 1: Visual percentage-wise context for the *SemanticAircraft* dataset showing dominant sky-context. Context across all four quadrants was merged.

	building	elevation	object	pavement	person	plant	sky	soil	vehicle	waterbody
Instances	**7.5**	**3.2**	**1.5**	15.8	**1.2**	**5.1**	57.2	6.3	0.9	1.3
Quadrants	4.2	2.8	1.2	**17.4**	1.1	4.0	**58.6**	**7.8**	**1.0**	**1.9**

Supervised: Instances and quadrants from *SemanticAircraft* with their respective domain labels are used for supervised classification with a CNN.

For the purpose of domain prediction, classification accuracy (recall) is the primary goal, although unsupervised mixture models used other metrics for parameters tuning. The parameters of the baseline algorithm were not tuned, instead were set once using human-expert knowledge. The dataset *SemanticAircraft* consistes of a set of 3854 instance and 13265 quadrant triplets: RGB input images, corresponding context vector $\mathbf{c}_{i/q}$, and ground truth (GT) domain label. After setting 20% of data aside for the test set, the experiment took place in two phases. In the first phase hyperparameters and architectural designs were tuned following the evaluation on the validation portion of the remaining 80% using method-specific metrics. For the second phase two separate versions of *SemanticAircraft* were used. The first consists of all remaining 20% of instances and quadrants. For the second all samples with the GT domain label other were removed. The final prediction results of phase 2 can be observed in table 2.

4.1 Baseline Threshold Model

Algorithm 3 proposed in this subsection serves as the baseline for domain prediction evaluation. The basic premise of the baseline was to develop an algorithm that works similar to human intuition: The relative pixel amount of every context-class contributes towards a certain domain-belief with a set strength if it is as-expected for any domain. For example, apron samples are commonly expected to feature vehicles while runway and sky are not. Observing the context for any patch, e.g. $\mathbf{c}_{vehicle} = 0.4$ meaning forty percent of pixels are vehicle, the ranges $\mathbf{r}$ of expected vehicle context for all three domains are checked and weights w added for every domain with bounds including 0.4. This cummulative score s signifies the level of distinction all context classes provides. This is done for every superclass with scores adding up to the domain score d. For classification, an image patch then has to reach a configurable score-threshold th. While simple to configure, this algorithm shows multiple shortcomings: 1) It is parameter-heavy, making tuning for a set of domains and extension to other domains difficult, 2) All parameters are partly dependent on expert-knowledge, informed by previous dataset-wide semantic context analysis, 3) Equal domain-scores lead to ambiguity – in this case, this ambiguity was solved with random tie-breaks 4) Patches not meeting the threshold signify high uncertainty in the

Algorithm 3 Thresholding domain prediction

Requisites: Set of context vectors $\mathcal{C}_{\mathbf{i/q}}$. Set of domains $\mathbf{d}$ and for every domain and superclass $\mathbf{s}$ consisting of classes c a certain range $\mathbf{r}_{x,y}$ and weight $\mathbf{w}_{x,y}$. Domain-prediction threshold of th and a decrease th_d.

function TDP($\mathcal{C}_{\mathbf{i/q}}, \mathcal{D}, \mathcal{S}, \mathcal{R}, \mathcal{W}, th, th_d$)
  for c in $\mathcal{C}$ **do** ▷ For every context vector
    d_s $\leftarrow \mathbf{0} : \mathbf{0} \in \mathbb{R}^{n \times 1}$ ▷ Initialize the domain score
    for d in $\mathcal{D}$ **do** ▷ And for every dataset
      for s in $\mathcal{S}$ **do** ▷ And superclass in that dataset
        s_s $\leftarrow \sum_i c_i, \forall c \in \mathbf{s}$ ▷ Aggregate context of all classes
        if s_s $\in [\mathbf{r}_{d,s,l}, \mathbf{r}_{d,s,u}]$ **then** ▷ Check if score is in range
          d_s$_d \leftarrow$ d_s$_d + \mathbf{w}_{d,s}$ ▷ Add a weight to the domain score
    if max(d_s) $> th$ **then** ▷ Take the top-1 domain
      $\mathbf{l_c} \leftarrow argmax$(d_s) ▷ And assign the domain label
    else
      $th \leftarrow th - th_d$ ▷ Or decrease threshold until domain is found
  return l ▷ Return domain labels for every image patch

context or out-of-context patches and it is unclear how this should be resolved. Despite these drawbacks, once set up for a set of domains and datasets, results are reproducible due to the deterministic nature and inference time is negligible.

4.2 Unsupervised Clustering and Mixture Models

The mathematical foundations of the unsupervised clustering and mixture models are detailed by [3]. It should be noted, that any created cluster are an internal mathematical construct and do not resemble the set of predefined domains. This makes interpretation in a classification setting not as straightforward as with CNNs. The scikit-learn python library [21] was used for implementation. The clustering model of choice was the variational Bayesian Gaussian mixture model (VBGMM), which provides a larger flexibility than popular K-Means or regular Gaussian mixture models. The optimal hyperparameters are those, where the silhouette-coefficient [15] is at a maximum in [0, 1]. Tuned parameters include the distribution prior (Dirichlet process vs. Dirichlet distribution), covariance type (full, diagonal etc.), initialization (K-Means, random) and number of active components to model the data. Parameters were tuned in a grid-search, the highest achieved coefficients are 0.702 and 0.766 for instances and quadrants respectively, only one cluster can be assigned at any time. It should be noted, that in some experiments, the number of clusters was not set, allowing the VBGMM to cluster the context vectors however it sees fit. This would result in up to 13 and 9 clusters for instances and quadrants respectively, a hint that the restraint to 3 or 4 clusters does not fully explain the distributions that created the context vectors. Finally, it should be noted hat context vectors were not assigned to any

of the 3 or 4 domains, but only to an equal number of clusters. This means the method is truly unsupervised at the cost of domain prediction accuracy. To obtain classification accuracy (recall), the best-performing permutation of possible cluster-assignments had to be found, since the clusters are not directly related to the defined set of domains. This was done by expressing the confusion matrix across clusters as a cost matrix and obtaining the minimization over all permutations of possible row/column combinations [18], 4! when including other, 3! otherwise. This yields the permutation of the confusion matrix with the highest diagonal sum.

4.3 Supervised Convolutional Neural Networks

For the CNN, not the set of context vectors are used as input data, but instead the images $\mathcal{I}_{\mathbf{RGB}}$ / $\mathcal{Q}_{\mathbf{RGB}}$ and corresponding class-labels $\mathcal{I}_{\mathbf{l}}$ / $\mathcal{Q}_{\mathbf{l}}$ obtained from domain annotation. The PyTorch library [20] was used for implementation. All samples were resized to 256x256 pixels to meet the input requirements. The tuned parameters included model-type and size (ResNet50 to ResNet34, ResNet18 and various DRN [33] and MobileNet [25] architectures), image augmentation strength, batch-size and the manual addition of a dropout layer. Parameters were tuned using 5-fold cross validation. The best performing models on both instances and quadrants turned out to be ResNet18 [13] variants. For both models, only light image-augmentation yielded the best results. A key difference between the models is the presence of dropout ($p = 0.5$) for the quadrants model. Thus the problem of model overfitting could be addressed by reducing model-size and adding a dropout layer to the architectures. With hyperparameter tuning of the models concluded, they can now be directly compared against another on the held-out test set.

4.4 Domain Prediction Results

Since this can be seen as a classification task, classification accuracy or recall was used. Table 2 shows the final obtained results.

It should be made clear that all three models serve different purposes and only quantifying their usefulness regarding inference accuracy does not give the full picture: While the CNN has given the best prediction performance, the requirement of annotations for the domains are a significant drawback, limiting its application in entirely new domains. Nevertheless, for an applied system that observes airplanes around airports the CNN model would be the preferred choice. For the baseline, the parameterization makes tuning and extension to other domains difficult. At the same time, since context statistics need to be analyzed for choices in data aggregation anyway (at least when dealing with a new dataset from the wild), the parameterization naturally evolves from these ideas and requires little more effort. While the variational Bayesian gaussian mixture model (VBGMM) performs the worst in terms of prediction accuracy, the insights the model can provide into the dataset structure, hinting at possible other subdomains besides apron, runway, sky and other, are noteworthy. If the number of

Table 2: Accuracy of all three models predicting domains of airplane instances and quadrants from *SemanticAircraft*.

	Instances		Quadrants	
	Including Other	Excluding Other	Including Other	Excluding Other
Baseline	0.588 ± 0.015	0.796 ± 0.011	0.639 ± 0.017	$\mathbf{0.799 \pm 0.006}$
VBGMM	0.586 ± 0.048	0.712 ± 0.06	0.539 ± 0.029	0.637 ± 0.083
ResNet18	$\mathbf{0.716 \pm 0.015}$	$\mathbf{0.854 \pm 0.011}$	$\mathbf{0.692 \pm 0.013}$	0.778 ± 0.006

clusters was not specified as it was the case in some VBGMM experiments, more than three or four clusters were created. The results indicate, that the limitation to apron, runway and sky was perhaps too strict. In future work, further analysis with clustering algorithms could provide important insights.

Thus, all three models have benefits and drawbacks but for the explicit task of domain prediction, the supervised CNN performed the best. The exclusion of any other samples does improve all model's performance, most significantly the baseline. The importance of filtering out-of-context samples is not apparent but it stands to reason that without the explicit context representation and filtering using context the CNN would perform worse. Finally the somewhat underwhelming accuracies can be broadly explained due the manual annotation of domain labels. Even for human experts the distinction between apron and runway was hard, especially for image quadrants, and the most common prediction errors was between these two domains. A more principled approach, perhaps using context statistics themselves to assign labels to image patches, could prove more fruitful.

5 Conclusion

With the proposed semantic context module context vectors were extracted from semantically segmented masks. These context vectors were used for improved data aggregation and domain prediction of images in the merged dataset *SemanticAircraft*. Images were further manually annotated with domain labels. Results show that all three domain prediction models, a novel baseline, unsupervised clustering model, and the CNN were capable of predicting domains with acceptable accuracy, although only inferences with the ResNet18 CNN are accurate enough to guide potential downstream models. For baseline and mixture models the fact that they do not require annotations is a significant benefit. The clustering method additionally provides data introspection. In future works, improved domain prediction results could be used to guide parameter-selection for downstream models fine-tuned on specific domains. Finally clusters created with the unsupervised models could be further analyzed for deeper insights into the visual context of the datasets.

References

1. Aditya, S., Yang, Y., Baral, C., Fermuller, C., Aloimonos, Y.: Visual commonsense for scene understanding using perception, semantic parsing and reasoning. In: Logical Formalizations of Commonsense Reasoning - Papers from the AAAI Spring Symposium, Technical Report. pp. 9–16. AAAI Spring Symposium - Technical Report, AI Access Foundation (2015)
2. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7) (2015)
3. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
4. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1209–1218 (2018)
5. Chen, X.: Context Driven Scene Understanding. Ph.D. thesis, University of Maryland (2015)
6. Chen, Y.H., Chen, W.Y., Chen, Y.T., Tsai, B.C., Frank Wang, Y.C., Sun, M.: No more discrimination: Cross city adaptation of road scene segmenters. In: 2017 IEEE International Conference on Computer Vision. pp. 2011–2020 (2017)
7. Choi, M.J., Torralba, A., Willsky, A.S.: Context models and out-of-context objects. Pattern Recognition Letters **33**(7), 853–862 (2012)
8. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: Nus-wide: A real-world web image database from national university of singapore. In: Proceedings of the ACM international conference on image and video retrieval. pp. 1–9 (2009)
9. Csurka, G.: A Comprehensive Survey on Domain Adaptation for Visual Applications. Springer (2017)
10. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**(2), 303–338 (2010)
11. Fu, J., Liu, J., Wang, Y., Li, Y., Bao, Y., Tang, J., Lu, H.: Adaptive context network for scene parsing. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6747–6756 (2019)
12. Geman, S., Bienenstock, E., Doursat, R.: Neural networks and the bias/variance dilemma. Neural computation **4**(1), 1–58 (1992)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
14. Huang, L., Peng, J., Zhang, R., Li, G., Lin, L.: Learning deep representations for semantic image parsing: a comprehensive overview. Frontiers of Computer Science **12** (08 2018). https://doi.org/10.1007/s11704-018-7195-8
15. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons (2009). https://doi.org/10.1002/9780470316801
16. Kendall, A.G.: Geometry and Uncertainty in Deep Learning for Computer Vision. Ph.D. thesis, University of Cambridge (2019)
17. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
18. Morbieu, S.: Accuracy: From classification to clustering evaluation (2019), [Online]. Available: https://smorbieu.gitlab.io/accuracy-from-classification-to-clustering-evaluation/ [Accessed: 29.05.2020].

19. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (2014)
20. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
21. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)
22. Rabiner, L., Juang, B.: An introduction to hidden markov models. ieee assp magazine **3**(1), 4–16 (1986)
23. Richardson, M., Domingos, P.: Markov logic networks. Machine learning **62**(1-2), 107–136 (2006)
24. Sakaridis, C., Dai, D., Van Gool, L.: Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7373–7382 (2019). https://doi.org/10.1109/ICCV.2019.00747
25. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4510–4520 (2018). https://doi.org/10.1109/CVPR.2018.00474
26. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. International Journal of Computer Vision **128**(2), 336–359 (Oct 2019), http://dx.doi.org/10.1007/s11263-019-01228-7
27. Sikirić, I., Brkić, K., Bevandić, P., Krešo, I., Krapac, J., Šegvić, S.: Traffic scene classification on a representation budget. IEEE Transactions on Intelligent Transportation Systems **21**(1), 336–345 (2020). https://doi.org/10.1109/TITS.2019.2891995
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2015)
29. Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., Xu, W.: Cnn-rnn: A unified framework for multi-label image classification. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2285–2294 (2016)
30. Wang, M., Deng, W.: Deep visual domain adaptation: A survey. Neurocomputing **312**, 135–153 (2018)
31. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. Journal of Big data **3**(1), 9 (2016)
32. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
33. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 636–644 (2017)

34. Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A.: Context encoding for semantic segmentation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 7151–7160 (2018)
35. Zhang, Y., David, P., Foroosh, H., Gong, B.: A curriculum domain adaptation approach to the semantic segmentation of urban scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**, 1823–1841 (2020)
36. Zhang, Y., David, P., Gong, B.: Curriculum domain adaptation for semantic segmentation of urban scenes. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2039–2049 (2017)
37. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision **127**(3), 302–321 (2016)