IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 July 2023, accepted 5 September 2023, date of publication 14 September 2023, date of current version 22 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3315589

RESEARCH ARTICLE

Feature Map Activation Analysis for Object Key-Point Detection

ALLEN RUSH, (Member, IEEE), AND SALLY WOOD, (Life Fellow, IEEE)
Department of Electrical and Computer Engineering, Santa Clara University, Santa Clara, CA 95053, USA
Corresponding author: Allen Rush (arush@scu.edu)

ABSTRACT Determining object location information in an image can enable more accurate CNN classification. Several extended CNN models have been developed to include both object location and classification into a unified model, at a cost of increasing compute complexity, increasing the number of parameters, and typically having a lower number of classification categories. We show that key-points within classifiable objects can be identified in early layer feature maps of a simple CNN without dependence on deeper layer processing or classification predictions. A statistical analysis of early feature maps is used to create a method for identifying and locating key-points that, with high probability, correspond to object locations in the image. This method uses only the forward pass of the simple CNN and requires no additional training. The method is tested on an image data set with known ground truth object locations as a function of the number of key points for four related selection methods. Results for object locations derived from key-points compare favorably to results obtained from R-CNN and are consistent over a range of key point set sizes.

INDEX TERMS Key-point, CNN feature maps, region proposals, kurtosis.

I. INTRODUCTION

In the past decade CNN models have been developed and improved from relatively simple architectures which accurately classify a single dominant object from a training set to more advanced models that include both object identification and location. Models such as VGG [1], Resnet [2] and Inception [3] can classify up to 1000 distinct object types, achieving classification performance of up to 94% for top five classifications for natural image scenes. Single purpose models such as VGG are widely recognized as very efficient and accurate for the simple classification of a single object.

Natural images often contain multiple objects of interest that may vary widely in size. Recent efforts have focused on identifying image regions, each of which attempts to isolate a single object. The use of regions in the form of bounding boxes can improve the prediction and reduce or eliminate ambiguity in the case of multiple objects in the image. Prior to 2012, earlier studies on object location included Selective Search [4], Histogram of Oriented Gradients [5], and Scale Invariant Feature Transforms [6]. These model-based approaches applied filters or kernels directly to the input images. More recent approaches use CNN backbones and network extensions to accomplish the more complex goal of object detection, localization and classification in a single network. Object detection was added to the classification process by identifying or estimating the object size and location in the training process. Combined with the classification task, this results in a system that can isolate and identify multiple unique objects within an image. Approaches such as YOLO [7], SSD [8], and Fast and Faster R-CNN [9], [10] are examples of models that combine these two tasks into a single model, at a cost of increasing the number of parameters, computational complexity, and training time. More recent advances based on extensions to R-CNN include [11], [12] which integrates a coordinate attention block in addition to Sparse R-CNN [13] to locate region proposals. In addition these complex models typically have a more limited classification variety than simple networks and are more computationally expensive to execute. Table 1 briefly summarizes the comparison between the simple VGG and R-CNN models. The inference pass for

The associate editor coordinating the review of this manuscript and approving it for publication was Yudong Zhang.

TABLE 1. Comparison between VGG and R-CNN models.

	VGG	R-CNN
Trainable parameters	134M	426M
Inference Computation	16Gflops	>300Gflops
Regions	Full image	300 to >2K
# Trained Objects	1000	20

VGG produces a confidence for classification identification for a single object, and the inference pass for R-CNN produces a confidence classification and location for one or more objects.

Considerable progress has been made in improving the detection and localization performance with recent models and approaches, but the performance of these models is still significantly lower than that of more simplified classification task. Liu et al. [14] summarized the current state-of-the art approaches for object detection. In Faster R-CNN for example, the region proposal function is a separate network within the model that evaluates the later convolution/activation layers to extract region proposals which define the local regions containing an object. Compared to simple classification models, a higher resolution input is required to produce reasonable proposals for multiple objects. Other models such as YOLO and SSD have different mechanisms for determining region proposals. They all require substantial compute complexity in multiple layers to obtain a final set of regions. A more recent advance in object localization is the G-CNN [15] in which bounding boxes are formed by starting with a multi-scale regular grid and iteratively improving the bounding boxes during training. The number of region proposals for the models mentioned above ranges from a few hundred up to 5,000 to 10,000 per image.

Object localization and isolation problems are more challenging when considering object scale, deformed objects and occlusions, and a first step to establish region proposals is to identify key-points of objects in an image. This study focuses on discovering key-points that can be used to form local partitions for cropping and subsequent classification using a simple CNN model. The objective of this study is to create a robust key-point identification method that does not require a complex CNN/object detection model or training. It uses a simpler CNN architecture that has been trained on ground truth data using network hyper-parameters appropriate for a specific application. In contrast to the region proposals described above, this approach leads to partitions that have a high probability of containing an object without additional training or the need for a separate region proposal network. Moreover, if we can select points from early feature maps, the computing cost is limited to only the first few layers of the network. These regions can be used to crop the input image such that each region can be processed individually. The classification confidence for each cropped image is used to validate the objects. We target images in which the number and size of objects match or are compatible with models such as R-CNN, YOLO or SSD. We use the VOC data set [16] for evaluating performance because it is often used in testing for object detection and classification, and it contains ground truth bounding boxes for multiple objects.

II. BACKGROUND

CNN models generally consist of a collection of convolution, activation and pooling layers followed by fully connected and classification layers. The feature extraction and encoding functions for the CNN model are executed in these hidden layers. In the training process, the convolution kernels are optimized to encode features in the convolution layers such that the final layer feature maps can be decoded to form a classification probability for the training set, typically through fully connected layers and softmax.

Following the notation of Khan et al. [17] let layer l be a convolution layer that processes K_{l-1} channels from the previous layer to produce K_l output channels. The output for the k^{th} channel in (1) is the sum of K_{l-1} convolutions using convolution kernel $e_l^k(u, v, c)$ for the c^{th} input channel.

$$f_{l+1}^k(p, q) = \sum_{c=1}^{K_{l-1}} \sum_{u=1}^{U_l} \sum_{v=1}^{V_l} f_l^c(p-u, q-v) e_l^k(u, v, c) \quad (1)$$

For $l = 1$ the input to the layer is the resized image where $f_1^c(p, q) = i(x, y, c)$ is an element of the input tensor $I \in \boldsymbol{R}^{X \times Y \times C}$. The input resolution of the model is (X, Y) and C is the image channel count, which is three for RGB images.

If layer l is an activation layer, a non-linear point processing activation function, which is typically tanh, RELU, or Leaky RELU is defined by

$$f_{l+1}^k(p, q) = g_a(f_l^k(p, q)) \quad (2)$$

where $g_a()$ is the activation function. A feature map is defined as the output of the activation layer.

The convolution and activation sequences in a deep model encode features from the input image. Features from the last feature map are input into one or more fully connected layers and decoded into a classification vector using a function such as softmax. Although variations on the basic CNN have been developed such as ResNet which uses residual layers [2] and Inception which uses a Network-in-Network approach [3], the principle of activation response still applies.

The receptive field of the feature maps in a given layer of a CNN model is determined by the size of the convolution kernel and the receptive field of the previous layers, which in turn is determined by the previous layer convolution kernel sizes and pooling operations. Pooling layers are used to decrease the feature map resolution and increase the receptive field relative to the size of the convolution kernel. Typically max pooling or average-pooling methods are used, often 2×2. In the early layers, before the first pooling layer, the feature maps are represented at the same spatial resolution as the resized input image, thus retaining a 1:1 correspondence between the (p, q) coordinates of the feature map and the center of its receptive field in the input image.

High activation values from the feature maps, which are strong responses of the associated convolution kernels, are most likely to propagate through pooling operations, and may survive to influence the final feature map layer. The evaluation of receptive fields and hidden layer information by Zhou et al. [18] established a connection between object identification and hidden layer features in deep CNNs. Further exploration of the use of local information to establish region proposals is described by Ding et al. [19]. It uses the results described in [20] which adds an optimized SIFT method to a CNN model to establish object key-points.

If key-points can be identified using only early layers of a CNN model, a foundation for evaluating early feature maps can be created to establish a prediction for object regions of interest using feature maps with small receptive fields. These regions can then be used for subsequent partitioning and cropping. When resized to the CNN input resolution, the cropped version of the original image is more likely to have a higher classification confidence in a simpler CNN if it isolates the object because the object will not be dominated by a stronger object in the input image.

Although the baseline CNN model was designed to produce a classification prediction at the output, several authors have investigated the meaning of the hidden layers in the network. Some approaches have been developed to analyze the relationship between individual feature maps and other layers in a CNN model and the classification output. This relationship provides a foundation for evaluating early feature maps to establish a prediction for object key-points. A key-point is defined as a unique or distinctive pixel or small region of an image that is invariant to rotation, scale and distortion. Similar to earlier key-point detectors such as SIFT [6] or SURF [21], key-points in feature maps are the result of training and the underlying training data set rather than being based on a general model of a key-point. Five significant methods for interpreting feature maps are Saliency Maps [22], [23], Layerwise Relevance Propagation [24], [25], Visual Attention [26], Activation Maximization [27] and Feature Importance Ranking [28].

Saliency maps, introduced by Law and Strother [22], define the difference in loss functions in a neural network. Simonyan et al. [23] generated saliency maps from CNNs that highlight the key areas of an object based on a prior prediction of a class. The saliency map is determined using back propagation and partial derivatives of the classification score with respect to the input image. Further improvements and variants of saliency maps, including GradCAM, have been described in [29], [30], [31], and [32]. An additional improvement in saliency detection was made by Kummerer et al. [33] in which a collection of feature maps from a trained network was functionally combined to determine the saliency location or gaze of an image.

In contrast, Layer-Wise Relevance Propagation introduced by Bach et al. [24], [25], [34] determines the importance of each pixel in an input image by back-propagating a measure of relevance from each intermediate layer and feature map. As in saliency maps, LRP also relies on a prior classification to start the backward propagation of the LRP values. Let R_j represent the relevance of the j^{th} neuron in one layer with J neurons and let R_k represent the relevance of the k^{th} neuron in the subsequent layer with K neurons. The value of the k^{th} neuron is a RELU operation

$$a_k = \max(0, \sum_{j=0}^{J} a_j w_{jk}) \tag{3}$$

where a_j is the activation value for the j^{th} neuron in the J^{th} layer with a_0 corresponding to the bias, and w_{jk} is the weight for neuron j contributing to neuron k. The weighted sum

$$R_j = \sum_{k=1}^{K} \frac{a_j w_{jk}}{\sum_{j'=0}^{J} a_{j'} w_{j'k}} R_k \tag{4}$$

determines the relevance value for any neuron in a layer.

Variations of LRP include *LRP-*ϵ, which adds a regularization constant, and *LRP-*γ:

$$LRP\text{-}\epsilon : R_j = \sum_{k} \frac{a_j w_{jk}}{\epsilon + \sum_{0,j'} a_{j'} w_{j'k}} R_k \tag{5}$$

$$LRP\text{-}\gamma : R_j = \sum_{k} \frac{a_j (w_{jk} + \gamma w_{jk}^{+})}{\sum_{0,j'} a_{j'} (w_{j'k} + \gamma w_{j'k}^{+})} R_k \tag{6}$$

The *LRP-*γ version is used to increase the relevance of neurons in early layers by increasing the influence of positive weights (γw_{jk}^{+}) This facilitates the final determination of the relevance of the input pixels and highlights the importance of identifying useful information in the early layers.

LRP provides a framework for associating neurons or feature elements from a feature map to the final class by back propagation of the class relevance. A similar relationship between the lower layers of a network and the final classification was described by Zhang et al. [26]. This approach is defined as a top-down attention process based on the probabilistic Winner-Take-All algorithm, WTA, which was initially developed as a selective tuning visual attention model [35]. In the probabilistic WTA model the back propagation process is described as an ''Excitation Backprop''. Let $P(a_j)$ be the Marginal Winning Probability for neuron a_j in the layer below the layer containing a_j. The Marginal Winning Probability (MWP) is given by

$$P(a_j) = \sum_{a_i \in \mathcal{P}_i} P(a_j|a_i) P(a_i) \tag{7}$$

where $\mathcal{P}_i$ is the parent node set of a_i. The conditional winning probability $P(a_j|a_i)$ is given by:

$$P(a_j|a_i) = \begin{cases} Z_i \hat{a}_j w_{ji} & \text{if } w_{ji} \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $\hat{a}_j$ represents the bottom up feature strength and is a positive value output of the activation function. The top

down feature expectancy is described by w_{ji}, and Z_i is the normalizing factor such that

$$\sum_{a_j \in C_i} P(a_j|a_i) = 1 \tag{9}$$

where $a_i \in C_i$ are the child neurons connected to a_j.

Finally, for Excitation Backprop, the authors introduced a variation called Contrastive Top-Down Attention. This is achieved by calculating another Marginal Winning Probability, constructed from the negative or complementary values of the weights. From (8) this would have the effect of selecting the remaining weights that are negative in the original MWP case. Subtracting the two probabilities removes any contribution from common winner neurons and amplifies the discriminative remaining winner neurons.

Although Saliency maps and LRP provide a linkage between output classification and input pixel contributions, they do not directly add to the understanding of how hidden layer feature maps are influenced by image input patterns and content. In [27] Erhan et al. defined Activation Maximization as a process for finding input patterns from an image that maximize an activation in a hidden layer. From this, we can conclude that the presence of high activation values in the hidden layers corresponds to input locations or regions that maximize or nearly maximize the response to the associated combination of kernels in the receptive field.

Related to Activation Maximization is the idea that features extracted in the convolution layers can be ranked according to their contribution to the classification task. Wojtas and Chen [28] describe a feature selector network that is jointly learned with a classification model to produce a ranking of features that correlates to the underlying classification prediction. This study was motivated by Random Forests [36] and dropout [37], and based in part on the earlier development of a framework for feature selection by Song et al. [38]. Extending the ideas of Feature Importance Ranking, the early layers of the network can contain features that indicate the presence and positions of key-points when ranked.

III. ANALYSIS OF EARLY FEATURE MAP LAYERS

Features from images processed by CNN models are extracted through layers of feature maps that encode spatial information which is ultimately used to determine a classification prediction and probability. Techniques such as Saliency Maps, LRP, Excitation Backprop and Feature Importance Ranking suggest that key-points can be identified within a feature map that can propagate to the last convolution/RELU layer. LRP and Saliency maps methods demonstrate a direct correspondence between high feature map response for a region that contains an object or portion of an object and the resulting classification prediction from the final layer in the model.

While the convolution layers and feature extraction for simple CNN models are mainly needed for object classification, our focus is on identifying the presence and location of objects in an image using information in the convolution layers and feature maps. The possibility of extracting information from early layers of a network to find object key-points in an image can be inferred from the LRP and Saliency results. In the LRP framework of [24], the authors noted that in *LRP-*γ (6), early feature maps are factored into the LRP equation by spreading the relevance of whole features rather than individual pixels. A second example from [27] shows that feature values can be maximized by matching the associated region of an image to the underlying kernels in a given feature map. We conclude from these previous studies that the results from every activation layer in a CNN model contain potentially useful information relating to the spatial extent and location of an object or key-points of an object, and it is possible that sufficient information may be obtained from a limited number of layers.

The success of transfer learning is another indication that the features extracted throughout the levels of a CNN are directly leveraged for a large class of objects beyond the original training set [39]. This would imply that the final classification is not needed to identify key-points in a trained network. In transfer learning the entire pre-trained feature extraction collection of convolution, activation and pooling layers is used without modification. Only the last few layers are trained for a new collection of classes with an associated training data set. This pre-trained backbone is robust to feature definitions even beyond the original training data set. Tammina [40], highlighted the contribution of early layer feature maps to transfer learning.

We postulate that the feature map information in the forward pass alone using a trained CNN model can be used to predict regions likely to contain an object or part of an object. Moreover, we propose identifying locations in early feature maps that are key-points of objects. We focus on the early feature maps without feedback from subsequent layers and on the relationship between feature map outputs and key-points that belong to significant objects. We expect the selection space from the feature maps to be sparse because relatively few selections from the feature maps identify unique characteristics or key-points of an object. These key-points can form the basis for constructing clusters and partitions for cropping and subsequent classification. As described by Menikdiwela et al. [41], a key idea in establishing a region of interest is to combine feature maps at each pooling layer with appropriate up-sampling. We extend this work to determine key-points in an early feature map layer that correspond to locations of objects in the input image. Moreover, clusters of key-points increase the likelihood that the associated region contains an object or a portion of an object that is a member of the collection of trained classes, as described by Lowe in [6]. Considering the success of transfer learning, a cluster of key-points may also indicate the presence of an object that is not a member of the trained classes.

With networks that use max pooling functions that preserve high activation values in subsequent layers, strong activations in the early layers of simple networks have a higher

probability of propagating through the network than weak activations. In the earlier layers, strong feature map outputs may be generated by some objects in the input image which, although they will not be selected in the final layers as the dominant objects in the image, may have been identified had they been the only object in the image. These points can be used to form clusters that correspond to localized areas containing objects that can be cropped from the original image and subsequently used as new inputs for a simple CNN. This increases the effectiveness of the simple CNN in several ways without retraining. Multiple objects may be identified rather than only a single object, and the relative location of multiple objects can be established. In addition the input resolution to the CNN may effectively be increased by the cropping operation

We use an approach similar to that of Ding et al. [19] in which key-points are used to establish anchor regions for training an R-CNN model. The prediction of key-points in our method is determined from early feature maps of a pre-trained simple CNN model as opposed to optimized SIFT [20]. We selected the VGG 16 model [1] for our simple CNN because it is widely used in research to study object classification and detection. It is a simple model, with only a single path through the network and consists of either 13 or 16 convolution layers. The input size is 224×224 pixels and larger images must be resized to a reduced spatial resolution.

The selected VGG 16 model was trained using the ILSVRC data set for 1000 classes [39]. This data set is representative of several common object classes and sub-classes in natural images. The earliest layer of interest is the deepest layer that has the same resolution as the input image, which, for VGG, is the RELU output following the second convolution layer. Identifying key-points in this layer enables us to form clusters with the same spatial resolution as the resized input image. It is also possible to consider activation layers deeper into the network, specifically layers immediately prior to the deeper pooling layers. The last activation layer prior to the fully connected layers used for classification has a resolution of $(X/2^{N_p}, Y/2^{N_p})$ where N_p is the number of 2×2 pooling layers in the network. As noted in [42], deeper layers in the CNN increase the receptive fields but decrease the positional resolution relative to the model input resolution. Using the deepest layer, key-point detection would produce predictions with a region ambiguity of $(x \pm 16, y \pm 16)$ pixels relative to the input image to the model, and the receptive field is 212×212, effectively the entire image. This introduces a substantial margin into the resulting bounding box estimates. In addition, in deeper layers, dominant objects may have already eliminated the key-points for other objects.

If the analysis of the activation layer before the first pooling layer does not produce an accurate set of key-points, selecting the activation layer before the second pooling layer would result in a margin for the bounding box prediction of $(x \pm 1, y \pm 1)$. This may be offset by the fact that deeper layers have increased receptive fields and may have stronger activations of key-point regions. We refer to the key-point regions in this case, because every layer after the first pooling layer has spatial ambiguity, which is a function of the number of pooling layers prior to the layer of interest.

FIGURE 1. Test images with ground truth bounding boxes.

A. ANALYSIS OF FEATURE MAP DISTRIBUTIONS

We begin by assessing the statistical characteristics of early feature maps to develop a method for selecting a small number of points from the collection of feature maps that have a high probability of being identified as key-points of objects. Two example test images in Figure 1, taken from the VOC test data set, demonstrate different arrangements of multiple objects. Image (a) is an example with multiple "boat" objects with bounding boxes that are not overlapping. The VGG model can only identify one of the three instances in the image. Image (b) shows an example of a test image in which there are multiple objects with overlapping bounding boxes. Key-points may be generated by surrounding objects as well as the dog.

For Figure 1(a) the top VGG classification is "boat" with a confidence of 0.99. The other four classifications in the top five, "dock", "seashore", "promontory" and "breakwater", all had confidence values less than 0.1. For this image VGG performs well with a high confidence classification, but there is ambiguity as to which instance of "boat" the classification prediction belongs and no indication that there are multiple objects. For R-CNN, there are three instances of detection of "boat", all with confidence greater than 0.99. In addition, the three estimated bounding boxes generated by the R-CNN are very close to the VOC ground truth bounding boxes.

In Figure 1(b) the top VGG classification is "dog" with a confidence of 0.95. The other four classifications in the top five, which all have a confidence of less than 0.05, are "doormat", "yurt", "sliding door" and "stove". The R-CNN detection is "dog" with a confidence of 0.998, and the estimated bounding box is very close to the ground truth bounding box. No other objects were identified by the R-CNN.

When considering the feature map information, where $f_l^k(p, q)$ is an element of the k^{th} feature map in layer l, we note that normalizing and centering the feature map distributions with $\mu = 0$ and $\sigma = 1$ will not be useful because the feature map values propagate through this model without normalization. However, the mean and central moments of

each feature map can be used to characterize the maps. For the k^{th} feature map the mean value $\mu_1(k)$ is defined by

$$\mu_1(k) = \frac{1}{PQ}\sum_{p=1}^{P}\sum_{q=1}^{Q} f_l^k(p,q) \tag{10}$$

and the n^{th} central moment is

$$\mu_n(k) = \frac{1}{PQ}\sum_{p=1}^{P}\sum_{q=1}^{Q} (f_l^k(p,q) - \mu_1(k))^n \tag{11}$$

The kurtosis of the k^{th} feature map, which is often used as an indicator of distributions with outliers, is defined in (12).

$$\kappa(k) = \frac{\mu_4(k)}{(\mu_2(k))^2} \tag{12}$$

High values of $f^k(p,q)$ in the k^{th} feature map are possible indicators of key-points of an object near location (p,q) in the image. Confidence that the high value is significant may be increased if the kurtosis of the map is high which would imply that the high values are sparse. The key property of kurtosis that we exploit is the relationship between high kurtosis and distributions with long tails or outliers which are expected for high activation response elements of the feature maps. For example, if the histogram of a feature map with outliers was approximately modeled very simply as a Bernoulli distribution, where the outliers can be assigned to a value of 1 with a probability of p and the non-outliers can be assigned a value of 0 with a probability of $(1-p)$, the central moments for the feature map distribution computed from (10) and (11) would be $\mu_2 = p(1-p)$ and $\mu_4 = p(1-p)(1-3p(1-p))$ and from (12), the kurtosis would be $\kappa = \frac{1-3p(1-p)}{p(1-p)}$. For a distribution with sparse outliers, the probability p would be very small and the approximate kurtosis would be $\kappa \approx \frac{1}{p}$, which would be large. Kurtosis values for individual feature maps vary widely from image to image, and depend on the response of the underlying kernels to the input image; however the outlier high values are likely to propagate to the next layer through max pooling.

Figure 2 shows four feature maps from the test image in Figure 1(a), which were selected from the collection of feature maps in the second feature map layer. Feature maps 55 and 51 have the lowest and highest maximum values, respectively. Feature maps 3 and 20 are the feature maps with the lowest and highest kurtosis values. The center column shows the four selected feature maps and the left column shows the output of the previous convolution layer before the activation layer. All images were scaled so that the lowest value is black and the highest value is white. The histogram of each feature map is shown in the right column with the horizontal scale set to show the range from zero to the maximum feature map value. The counts in the histogram bins were displayed using a logarithmic scale so that outliers are more visible.

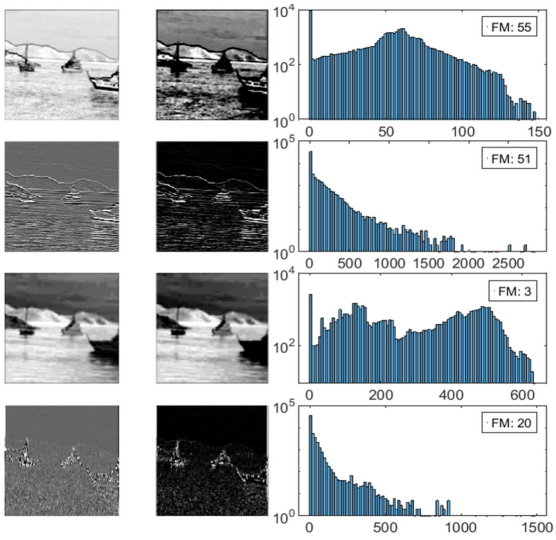

FIGURE 2. Four feature maps from Figure 1(a).

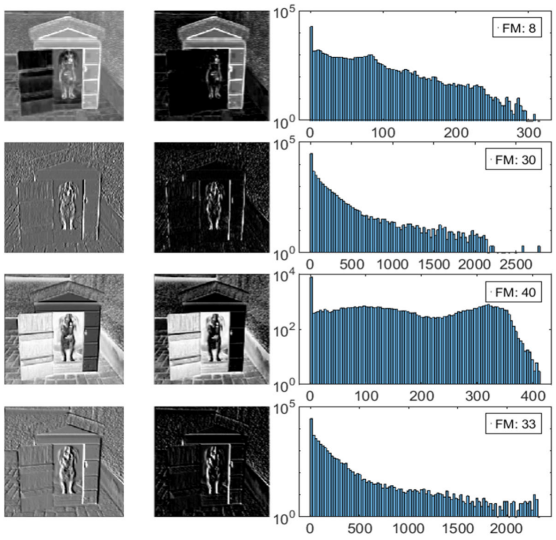

FIGURE 3. Four feature maps from Figure 1(b).

Significant differences are observed in these four feature maps. In rows 1 and 3 of Figure 2, the feature maps have low maximum values and low kurtosis values. A large number of values close to the maximum value can be observed, and visual inspection shows that almost all of the relatively high values correspond to background areas, not to boat objects. In contrast, in rows 2 and 4, the feature maps have high maximum values and high kurtosis values. These feature maps contain very few values near the maximum value, and the images of the feature maps are much darker than those of the other two rows. The bright points in these two feature maps are concentrated in the areas of the three boat objects.

A similarly selected set of four feature maps from the test image shown in Figure 1(b) is shown in Figure 3. Again, the feature maps that have a high maximum value and a

TABLE 2. Summary statistics for figures 2 and 3.

Selected Feature Map Statistics - Figure 2						
Feature Map k_i	Max Hist Value	Max $f(p,q,k_i)$	Mean	Var	Skew	κ
55	9307	147	47	30	-0.2	2
51	28992	2836	65	148	5.4	52
3	2780	638	287	176	0	1.5
20	28734	1482	19	50	9.1	143
Selected Feature Map Statistics - Figure 3						
8	19590	315	38	48	1.6	6
30	28712	2782	76	183	5.6	47
40	8181	413	155	119	0.2	1.7
33	27150	2337	73	172	6.2	56

high kurtosis value have bright points concentrated on the dog object and on edges in the surrounding structure. The specific maps that produce the high maximum values and high kurtosis values depend on the image content, and the maps selected in Figure 3 are not the same as those in Figure 2.

Table 2 summarizes the statistics for the four feature maps in Figures 2 and 3. The column labeled Max Hist Value shows the number of feature map values in the most populous bin, which is also the first bin, of each histogram. There is no close relationship between the mean and maximum values or between the variance and kurtosis. For the feature map distributions in rows 2 and 4 of Figures 2 and 3, high activation values and outliers are present. These high values, which are also outliers, have a higher probability of being selected in the max pooling operation that follows the feature map layer. Therefore, it is reasonable to assume that many of these points will be associated with objects that are identifiable by the trained VGG. This type of distribution is well represented by a distribution that has high kurtosis or is leptokurtic. When this exists, we can create a selection strategy for the feature maps guided by the kurtosis value of the distribution.

B. KEY-POINT IDENTIFICATION

We augment the definition of a key-point of an object in an image as a point that has some unique feature or attribute in the image that produces a high activation in one or more feature maps. For any given key-point in an image, one or more feature maps may reflect this uniqueness in the form of any high activation response which is an outlier relative to the other responses in the feature map. Each feature map kernel that produces a high response to key-points was trained to extract features from the collection of training data that correspond to these key-point locations. Moreover, as shown in [28], features that qualify as key-points often have a higher importance relative to other features. By applying the reverse logic of Saliency Maps and LRP, features which have high activation values and that are also highly differentiated are more likely to propagate to the final convolution layer prior to decoding as a classification prediction. There is high variability in the feature map content, which is a function of the underlying input image content, and the diversity of the

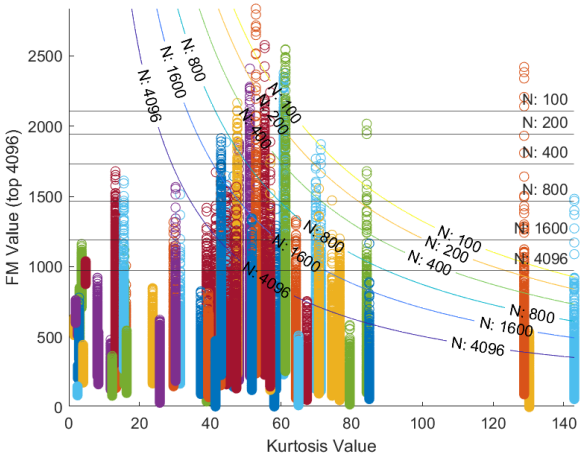

FIGURE 4. Top 4096 values from each feature map, vs feature map kurtosis, from Figure 1(a).

feature map responses allows a wide range of different object types to generate sparse high value responses. These two properties of activation values and key-point identification lead to the definition of a target density profile for the histograms of each feature map.

In the opposite case, in areas of the image in which no or very low activation response is observed, the density and number of these low values can be very high. In many cases they are tightly grouped near zero. A third case is one in which there are large areas of relatively high response. This might be an example of large areas of background such as sky or low-frequency patches in an image that are not sparse. These responses offer no practical information regarding the object key-points.

C. SELECTION OF FEATURE POINTS

Without the assistance of Saliency mapping or LRP, we can establish the key-point contribution potential based on feature map values and statistical measurements. The goal is to identify key-points and distinguish them from other, potentially high activation value, feature map locations. An initial approach may be to select the highest activation values from each individual feature map. However, the highest values in one feature map may be low compared to other feature maps, or may correspond to low interest or background areas of an image. For 64 224 $\times$ 224 feature maps, there will be a total of approximately 3.2 million points, and the number of key-points should be a small fraction of these points. Rather than selecting the highest value points from each feature map as potential key-points in the layer, it is more useful to consider only the top N points from a combination of feature maps to find the higher values that will survive the max pooling function. In addition, as Figures 2 and 3 demonstrate, there is a wide range of maximum individual feature map values, and not all locally high values are associated with objects; therefore for any specific image, some feature maps identify more key-points than others.

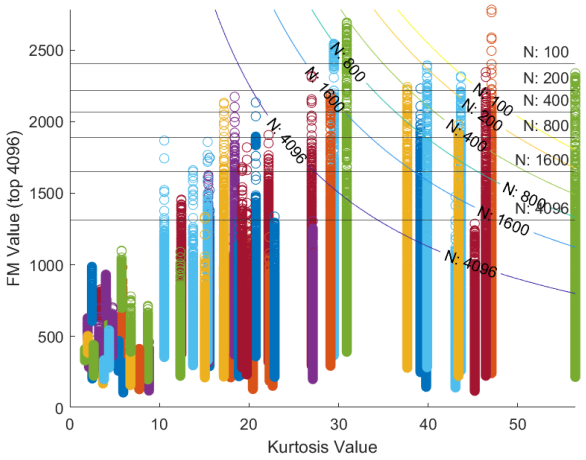

FIGURE 5. Top 4096 values from each feature map, vs feature map kurtosis from Figure 1(b) showing larger concentration of high kurtosis feature maps.

Let S_N be a relatively small selection of N points from a combination of feature maps. The effects of two different strategies for selecting S_N are demonstrated in Figures 4 and 5. In Figure 4 all the top feature map values for Figure 1(a) are plotted, vertically, and aligned horizontally with the kurtosis value of the feature map. For clarity, only the highest 4096 values from each feature map are displayed. Horizontal lines show the global threshold value which will select the N highest value points from all the feature map values $f^k(p, q)$. For N=100, only six of the 64 feature maps contribute to S_N. As N is increased, more feature maps contribute to S_N, but at the largest value shown, N=4096, there are still many feature maps that do not contribute to S_N.

A second selection strategy prioritizes high kurtosis by selecting the N points with the highest values of $\kappa(k) \cdot f^k(p, q)$, the product of the feature map value and the feature map kurtosis. The contour lines represent isolines to indicate the thresholds of constant selection size with kurtosis weighting for each feature map. When weighted by kurtosis, most of the top 100 values come from two feature maps with kurtosis values of 129 and 142 and the rest come from only a few other feature maps with kurtosis values between 50 and 60. Similar responses are observed with higher values of N. With both threshold definitions, when N=4096, contributions come from the majority of the feature maps. In the weighted kurtosis case, feature maps with very low kurtosis values less than 35 do not contribute to any points. There is some overlap in the sets of points selected by the two methods.

A second example of feature map values and selection strategy, based on the test image in Figure 1(b), is shown in Figure 5. In this case, a larger number of feature maps have a relatively high kurtosis, which results in a larger number of different feature maps contributing their high value points to S_N. This is an indication that the image may contain different object types. Compared to the image with three similar objects where the same feature maps probably respond strongly to all three objects, feature maps responding strongly to the dog and feature maps responding strongly to the surrounding structures are different, so more feature maps will have high kurtosis values.

D. KEY-POINT SELECTION ACCURACY

The accuracy of a specific key-point selection will be evaluated using VOC ground truth bounding boxes. This establishes the key-point contribution potential for each individual feature map. Feature map statistics are analyzed by first evaluating the selections of each feature map in the selected layer. Each point in the k^{th} feature map is identified as a ground truth positive if it is included in a ground truth bounding box or a ground truth negative if it is not included. Since we want to locate all significant objects in an image we considered the union of all ground truth bounding boxes in an image as the collective ground truth. A selection S_N^k, which includes the top N values of the k^{th} feature map, represents a set of points assumed to be key-points. This set can be evaluated based on how many points are ground truth positive, or true positive. Let the subset of points that lie inside a ground truth bounding box be defined as $S_{N_P}^k$, the true positives, and the subset of points that lie outside all ground truth bounding boxes be defined as $S_{N_N}^k$, the false positives. Then $||S_{N_P}^k||_0$ is the number of points inside the ground truth bounding boxes and $||S_{N_N}^k||_0$ is the number of points outside the ground truth bounding boxes.

The precision or positive predictive value of the S_N points from the k^{th} feature map is

$$PPV_N^k = \frac{||S_{N_P}^k||_0}{N} \tag{13}$$

where $N = ||S_{N_P}^k||_0 + ||S_{N_N}^k||_0$. A precision value near 1 indicates that almost all the selected points lie inside the ground truth bounding boxes. Because it is presumed that key-points will be used to form clusters, and bounding boxes of clusters will define the proposed partitions, it is not necessary that S_N include all the points in the true bounding boxes, and sensitivity is not a relevant measure of performance. Higher values for precision or PPV indicate a high probability that clusters will identify areas within the known object bounding boxes.

For the four feature maps from Figure 2, Figure 6 shows the precision of each as a function of N for $1 \leq N \leq 4096$. Similarly, Figure 7 shows the precision for the feature maps from Figure 3. In general, the feature maps with low maximum values or low kurtosis in rows 1 and 3 have lower precision compared to feature maps with high maximum value or high kurtosis. In Figure 6 the precision in rows 1 and 3 is less than 50% for N>200 but in rows 2 and 4 the precision is greater than 50% for N<2000. In Figure 7 precision values are lower compared to Figure 6 because the visually significant structures surrounding the dog are not identified as objects by VOC. However, as in Figure 6, rows 2 and 4 show significantly better performance

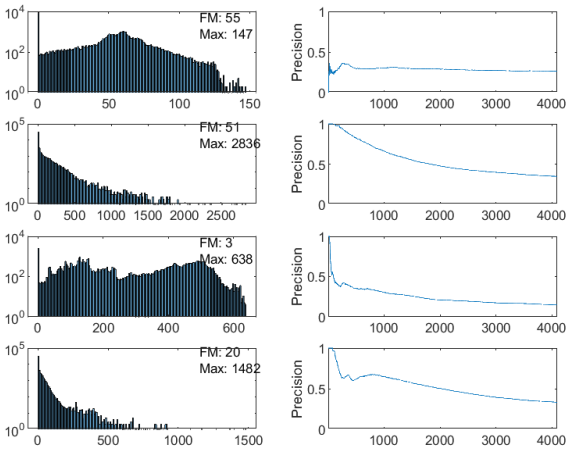

FIGURE 6. Precision as a function of selection set size for Figure 2.

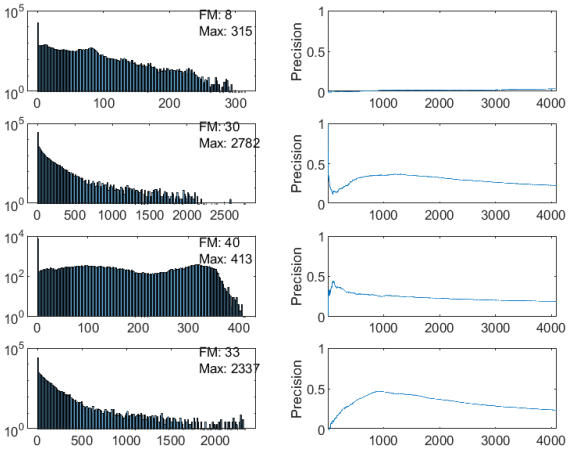

FIGURE 7. Precision as a function of selection set for Figure 3.

than rows 1 and 3. At N=2000, S_{2000_P} has 800 and 900 points in rows 2 and 4, respectively, compared to 10 and 500 in rows 1 and 3.

When selections of S_N are made using these approaches, the points contributed from each feature map are not necessarily unique (p, q) points. It is possible and even likely that the S_N sets from multiple feature maps contain selections for the same (p, q) location. We also note that isolated key-points alone will not be sufficient to generate a classification response. Local clustering of key-points from one or more feature maps is a strong indication of the presence of an object. This leads to the recognition that certain features are more important than others, as described in [28] and [43]. The feature ranking process suggests a statistical distribution that correlates to a measure of the sparseness of highly important features. When the number of key-points that correspond to high ranking features of distinguishable objects is a small fraction of the total number of pixels, this is a reasonable assumption. Thus, the individual feature maps should be combined such that the feature ranking property is embodied in the resulting combination.

IV. GLOBAL FEATURE MAP CONSTRUCTION AND PERFORMANCE

A unified Global Feature Map can be defined with ranking property determined by a function of the K feature maps in the a feature map layer, $f : \boldsymbol{R}^{P\times Q\times K} \rightarrow \boldsymbol{R}^{P\times Q}$. We compare four approaches: feature map sum, GFM_{sum}, feature map max, GFM_{max}, kurtosis-weighted sum, $GFM\kappa$ and kurtosis-weighted max, $GFM\kappa_{max}$. The $GFM_{sum}(p, q)$ is the sum of all feature map values at point (p, q). This creates a high value when multiple feature maps have a high response at the same point. It is defined as

$$GFM_{sum}(p, q) = \sum_{k=1}^{K} f^k(p, q) \tag{14}$$

The $GFM_{max}(p, q)$ is the maximum value of all the feature maps at point (p, q). Since this has only the highest value for each (p, q), it does not benefit from reinforcement of high values from other feature maps at the same location. It is defined as

$$GFM_{max}(p, q) = \max_k(f^k(p, q)) \tag{15}$$

Two other GFM maps are defined in a similar way but with feature map values weighted by feature map kurtosis to additionally emphasize feature maps with sparse distributions. The $GFM\kappa(p, q)$ is the kurtosis weighted sum of $f^k(p, q)$ from all feature maps. This will increase the contributions of outliers which will increase the likelihood of selecting key-points. It will reduce the relative ranking of (p, q) values that are not outliers and are more likely the background in the image, as observed in Figures 6 and 7. Higher-order weighting functions using κ^r would have the effect of increasing the contribution of the highest kurtosis feature maps and reducing the contribution of others, and for very large r, only the feature map with the largest kurtosis would be used for point selection. High values of r would be undesirable because in an image with multiple objects of different types, it is expected that different feature maps would have a strong response for different types of objects. $GFM\kappa$ is defined as

$$GFM\kappa(p, q) = \sum_{k=1}^{K} \kappa(k) \cdot f^k(p, q) \tag{16}$$

The $GFM\kappa_{max}(p, q)$ is a modified version of GFM_{max}. Each value $f^k(p, q)$ in the feature map is multiplied by the kurtosis value of the feature map before the maximum is selected. The $GFM\kappa_{max}(p, q)$ is defined as

$$GFM\kappa_{max}(p, q) = \max_k(\kappa(k) \cdot f^k(p, q)) \tag{17}$$

Figures 8 and 9 show the resulting four types of GFMs as gray scale images for the test images in Figure 1(a) and (b). The highest values from the GFMs are bright and show the locations of potential key-points. Based on visual inspection

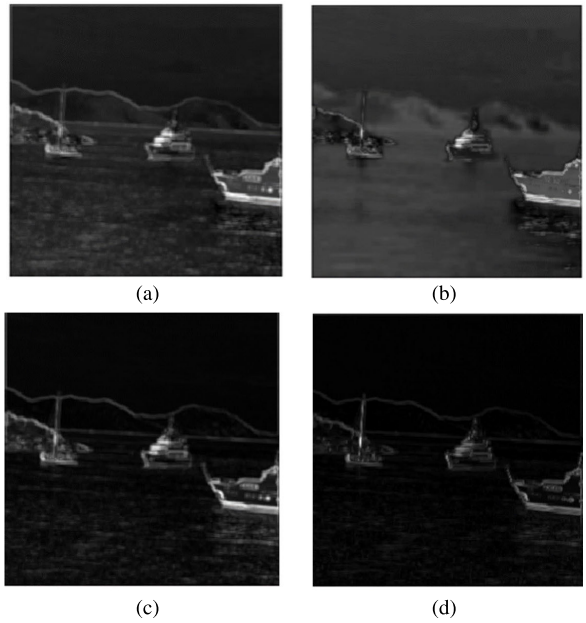

FIGURE 8. Images of GFM types for Figure 1(a).

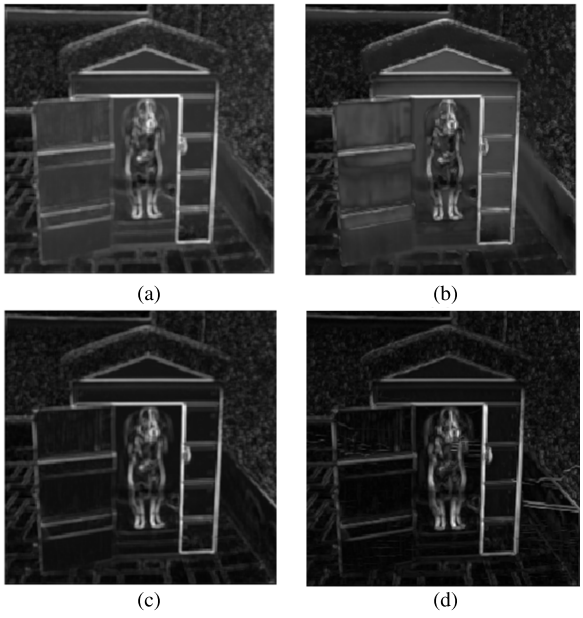

FIGURE 9. Images of GFM types for Figure 1(b).

of Figure 8, the GFM_{sum} and $GFM\kappa$ images show light points of approximately equal value from the boat objects that are much brighter than the rest of the image. The two images of GFMs on the right that are based GFM_{max} and $GFM\kappa_{max}$ have fewer very bright points although there are moderately bright points along the edges of the boats. The images in Figure 9 have similar properties although there is less variability across the four GFMs. Bright areas correspond to both the ''dog'' and the surrounding structure. The kurtosis weighted GFMs in Figures 9(c) and 9(d) show less response to the surrounding structure than the other two

FIGURE 10. Test image (a) top selections using $GFM\kappa$.

FIGURE 11. Test image (b) top selections using $GFM\kappa$.

GFMs. Figures 10 and 11 visually show the locations of the highest GFM points in red for specific selection set sizes. When the top 100 values of $GFM\kappa$ are selected in Figure 10 they all fall within the bounding boxes of the boats. As the selection size increases to 800 there is a denser distribution of points in the boat object areas and a few points are found on part of the landscape background. In Figure 11, for N=400, selected points are clustered along the edges of the structure and distributed around the dog. When N=1600 the cluster of points around the dog is denser and a few points are clustered along additional edges in the structure.

Although these visual examples demonstrate the potential of selecting feature points from a GFM, a performance metric is needed to compare the different GFMs and evaluate how the value of N affects performance. The four GFM methods were evaluated based on their precision measure as a function of N, the number of selected points. Figure 12 shows a plot of PPV_N for the four GFM methods for the two test images as a function of N, with test image (a) and on the left and test image (b) on the right. For all four GFMs of the test image in Figure 1(a), for low values of N, most of the highest feature map values were part of one of the three boat objects as visually evident in Figures 8 and 10. As N increased the precision decreased slowly to $70\tilde{}\%$ at N=2000 and $60\tilde{}\%$ at N=4096. All four GFMs show a similar performance although $GFM\kappa$ has a slightly higher precision than the other GFMs.

The four GFMs for the image in Figure 1(b) show a different trajectory. The structures around the dog in Figure 1(b) are visually significant. For small values of N the precision is low because so many of the selected points are clustered along the edges of the structure surrounding the dog, and although the structure was not identified by either VGG or R-CNN, it might have been identified by different training data. For this reason, the precision in

Figure 12 is much lower for the test image in Figure 1(b), and it increases as N increases, up to $N \approx 2000$, which in this case proportionately adds more "dog" key-points than "structure" key-points. For N between 2000 and 4096 the precision is approximately 30%. Using VGG on images cropped based on clusters of the key-points, the classification confidence would be used to accept clusters around the "dog" and reject most of the clusters associated with the "structure". For this image $GFM\kappa_{max}$ has the highest precision and $GFM\kappa$ is similar but slightly lower. Note that even though there is a large difference in precision between the two images, the selection measurements still find significant object key-points even with relatively low precision, as is the case with the test image (b). However, for any regions defined by clusters of only structure points, the classification confidence for the cropped image would be low.

As discussed in Section III layers other than than layer 5, the last activation layer before the first pooling layer, could be considered. Figures 13 and 14 show the values of $||S_{N_P}||_0$ for the four types of GFM and compare GFM methods for layers 3, 5 and 10 for test images. Layer 3 is the first activation layer, and layer 10 is the last activation layer before the second pooling layer. In addition, for reference, results for the simple magnitude of the gradient computed from the two directional Sobel filters are shown with a dashed green line. The lower dashed line indicates expected results from a random selection of points, and the upper dashed line indicates the perfect results if $||S_{N_P}||_0 = N$. In both test images, the selection of layer 5 produces higher values of $||S_{N_P}||_0$ and precision compared to either of the other two layers. In Figure 13 the results for layer 3 are similar to the simple Sobel gradient magnitude and the results for layer 10 are worse than the Sobel results. The layer 5 results show that all methods have approximately the same performance for N less than 2000, and all GFM methods are a substantial improvement over the Sobel filter when N is larger than 2000. In Figure 14 all methods show lower values of $||S_{N_P}||_0$ compared to Figure 13 because the visually significant background structures, which are not identified as a ground truth object, also produce points in S_N. The Sobel reference results have the highest value of $||S_{N_N}||_0$ for this reason. The GFM methods outperform the Sobel results for all cases and the layer 5 results for all GFM methods outperform layer 3 and layer 10. Based on the layer comparison results for the two images, the evaluation of key-point prediction for the larger test sets will focus on layer 5.

When precision is high and most of the selected points belong to an object, points added to the selection set as N increases which are not inside the object bounding boxes are not necessarily problematic for a number of reasons. They may be widely scattered and may not form clusters. They may be close to an object bounding box and would slightly augment the size. They may cluster around another object that is not part of the class collection identified in the ground truth data. In all cases, clusters associated with objects or spaces that are not part of the classification set will have low classification confidence after cropping and processing and will thus be rejected.

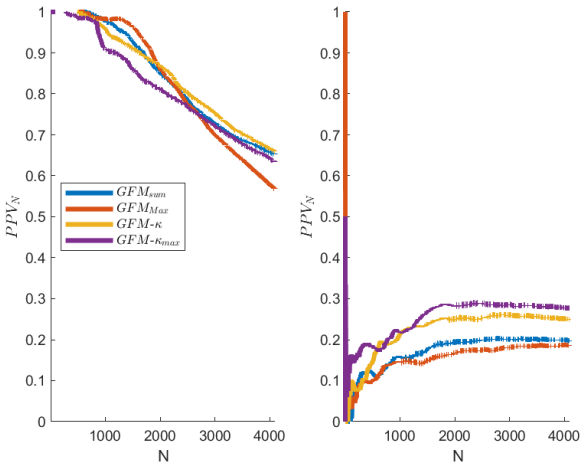

FIGURE 12. PPV_N of GFM methods for the images in Figure 1(a) , (b).

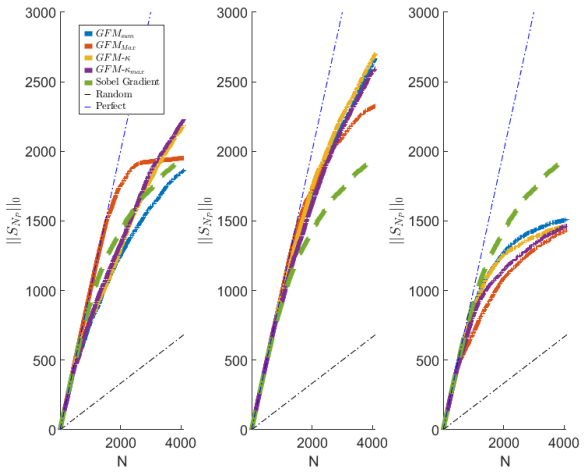

FIGURE 13. $||S_{N_P}||_0$ of GFM methods for the images in Figure 1(a).

A. KEY-POINT PREDICTION RESULTS

The PPV analysis can be used to evaluate how well the GFM methods perform over a large image test set. The initial assessment of the feature map selection process was performed using a subset of the VOC 2007 test images. While the VGG model was trained using the ILSVRC, as noted in the earlier transfer learning discussion, the pre-trained feature extraction layers are suitable for other object types such as those in the VOC test set. A variety of images were included in the test set, covering object scale, number of objects, and number of object types. In order to explore any dependence of precision on image content we consider ground truth bounding boxes from the VOC data set for four subsets of classes: "boat", "cat", "car" and "dog". The number of test images I_t is 200 for each of the four selected classes. The results were measured against the VOC

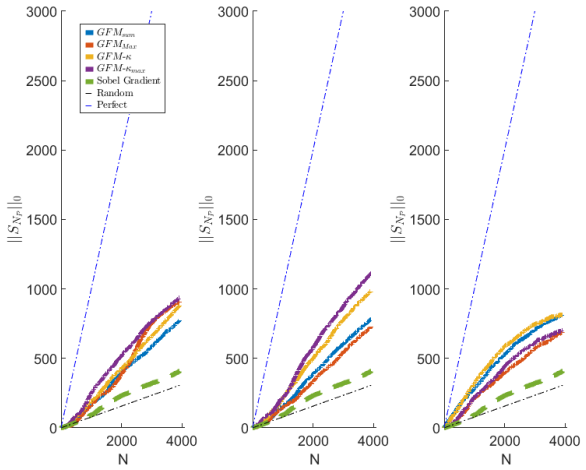

FIGURE 14. $||S_{N_P}||_0$ of GFM methods for the images in Figure 1(b).

TABLE 3. Statistics of ground truth bounding boxes.

Ground Truth Bounding Box Statistics							
Class	Num BBs	Max BB/ Image	Mean BB/ Image	Max BB Size	Min BB Size	Mean BB Size	Med BB Size
Boat	681	20	3.4	99.6	0.02	10.3	1.9
Cat	315	6	1.6	99.6	0.32	41.8	36.1
Car	761	21	3.8	99.6	0.03	10.8	2.5
Dog	492	18	2.5	99.1	0.08	25.8	16

ground truth bounding boxes with the selection range of N between 100 and 4096. The lowest median relative object size from the VOC test data set is 2% or approximately 32×32 pixels which corresponds to an upper limit of about 50 object bounding boxes with no overlap. The mean size of bounding boxes ranges from 11% to 46%, corresponding to a maximum of 9 and 2 non-overlapping object bounding boxes. Table 3 shows the statistics for the bounding boxes in the test image collection. The mean number of ground truth bounding boxes ranges from 1.6 to 3.8 while the maximum ranges from 6 to 21. Bounding box sizes are shown as a percentage of the input image size and range from 0.02% to 99.6%, with the mean ranging from 10.3% to 41.8% and the median ranging from 1.9% to 36.1%. Note that there is a large difference between the mean and median bounding box size for the "boat" and "car" classes compared to the "cat" and "dog" classes.

The performance of each class was reported as a vector of the sorted precision values obtained over the number of test images in the test set. The precision value for each image $i \in I_t$ as a function of N is given as

$$PPV_N^i = \frac{||S_{N_P}^i||_0}{N} \tag{18}$$

and average precision, AP, for all of the test images in a set is defined as

$$AP_N = \frac{1}{I_t} \sum_{i=1}^{I_t} PPV_N^i \tag{19}$$

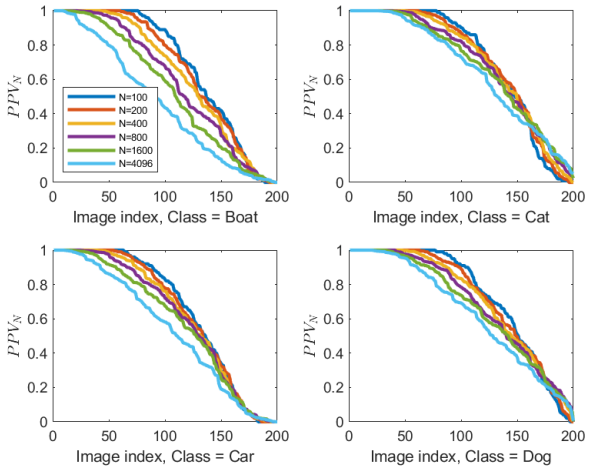

FIGURE 15. Key-Point PPV_N using GFM_{max} from test images.

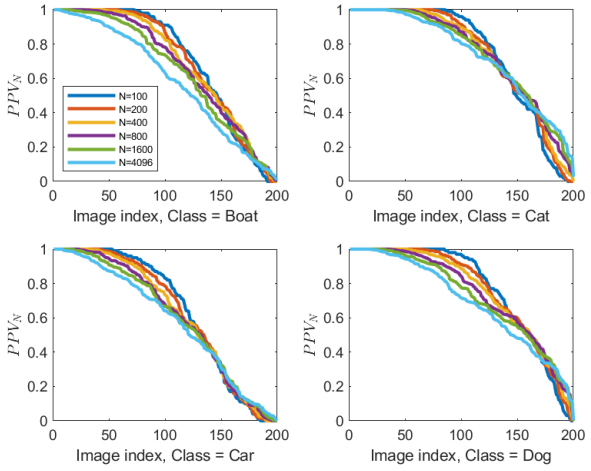

FIGURE 16. Key-Point PPV_N using $GFM\kappa$ from test images.

for each of the four test classes and for all four GFM methods.

Figures 15 and 16 show the sorted PPV_N results for the four classes over the selection range using GFM_{max} and $GFM\kappa$. The $GFM\kappa$ results in Figure 16 are very similar to the results for GFM_{sum} and $GFM\kappa_{max}$, whoch are not shown in plots, and these three GFMs show better performance that GFM_{max} in Figure 15. For GFM_{max} the precision is substantially lower for large N compared to $GFM\kappa$ and reflects the results in Figures 8 and 9.

In all cases there is little dependence of precision on N over the range of 100 to 1600. When N is increased to 4096, the precision values are somewhat lower as discussed earlier for individual feature maps. The four sub-classes of images based on content also show similar behavior with the largest variation shown in the "boat" set using GFM_{max} in Figure 15. For all four image classes half of the images have a PPV_N greater than 0.5. Visual examination of the images with the lowest PPV_N shows that most have objects with very small bounding boxes.

TABLE 4. AP for four image classes.

AP_N for GFM_{sum}						
N	100	200	400	800	1600	4096
Boat	70.1	68.9	67.4	65.8	63.3	57.5
Cat	74.6	74.2	73.6	73.1	72.6	70.7
Car	64.4	63.4	62.1	61.0	60.3	58.6
Dog	75.1	74.5	73.6	72.4	70.2	67.3
AP_N for GFM_{max}						
Boat	68.5	66.3	63.4	59.2	54.5	46.0
Cat	71.2	71.2	70.7	69.7	68.5	65.8
Car	65.7	64.6	63.0	61.6	59.2	53.8
Dog	73.6	72.4	71.1	69.2	67.3	63.7
AP_N for $GFM\kappa$						
Boat	71.0	70.3	69.5	67.5	64.8	59.3
Cat	74.4	74.3	74.5	74.6	74.5	73.6
Car	64.8	63.7	62.5	61.3	60.6	58.6
Dog	77.0	76.2	75.6	74.3	72.3	69.3
AP_N for $GFM\kappa_{max}$						
Boat	70.8	69.8	68.6	66.5	63.7	58.2
Cat	72.1	72.3	72.3	72.7	73.0	72.7
Car	64.3	63.5	62.6	61.8	60.5	58.0
Dog	72.4	72.7	73.1	72.4	71.0	68.6

TABLE 5. AP for single object detection.

AP_N for $GFM\kappa$								
N	100	200	400	800	1600	4096	Mean BB	Med BB
Boat	71.0	70.3	69.5	67.5	64.8	59.3	10.3	1.9
Cat	74.4	74.3	74.5	74.6	74.5	73.6	41.8	36.2
Car	64.8	63.7	62.5	61.3	60.6	58.6	10.8	2.5
Dog	77.0	76.2	75.6	74.3	72.3	69.3	25.8	16.0
AP_N, Single Object, $GFM\kappa$								
Boat	63.3	62.6	61.6	59.7	57.9	53.4	26.4	22.6
Cat	70.1	69.9	69.5	69.1	68.7	67.0	58.7	59.6
Car	69.8	69.2	68.4	67.8	67.0	65.0	33.0	26.2
Dog	71.3	70.6	69.8	68.9	67.7	65.3	48.1	51.6

A clearer distinction between GFM methods is apparent when the AP_N results for the four GFM methods are summarized in Table 4. The average precision, for selected values of N, is shown for all four GFM types and for all four image test sets. Although some of the differences are small, the $GFM\kappa$ method has a higher AP for almost all values of N and for all object classes and the GFM_{max} method has the lowest. The exception is the ''car'' set of test images, in which the results using $GFM\kappa_{max}$ are 0.1 to 0.8% greater than those using $GFM\kappa$ for N=200, 400, 800, and 1600. Table 4 shows that the key-point average precision has little variation with respect to the value of N between N=400 to N=1600.

The effect of multiple objects can be explored by comparing the results in Table 4 to results for subsets of each class in which images contain only one object. Table 5 shows the performance of key-point selection in test images in which there is only a single ground truth object, using $GFM\kappa$. The AP range for images with single objects using $GFM\kappa$ is 53.4 to 71.3, compared to 59.3 to 77.0 for all images in a set, as shown in Table 4. While the AP for ''boat'', ''cat'' and ''dog'' classes are lower in the single object results, the ''car'' results are higher. This is due in part to the difference in median bounding box size which is 2.5% for multiple object images compared to 26.2% for single object images. In addition, from Table 5, the mean for single object bounding boxes is considerably higher than the mean for single or multiple object bounding boxes. For multiple object images, there is a large difference between the mean and median bounding box sizes for ''boat'' and ''car'' classes compared to the ''cat'' and ''dog'' classes, but for single object bounding boxes, the differences between the mean and median are smaller.

B. KEY-POINT PARTITION IOU MEASURES

When clusters are generated from key-points, using well established clustering methods such as K-Means [44], [45], they form the basis for generating partitions that define an estimated bounding box for cropping and subsequent classification by the whole network. Although the AP results from layer 5 in the previous section show that the GFMs find points that with good probability will be included in an object bounding box over a wide range of N, it is also necessary that the selected points be distributed within the bounding boxes so that clusters of them will cover or nearly cover the full area of the objects' bounding boxes.

An estimated bounding box from the collection of S_{N_P} points can be used to compute the Intersection over Union, IOU, relative to the associated ground truth bounding box. For the i^{th} ground truth bounding box, B_i, the coverage by S_{N_P} will be estimated based on the subset $S_{N_{P_i}}$ of points in S_{N_P} that are located in B_i. Let the estimate $\hat{B}_i$ be the bounding box for $S_{N_{P_i}}$. Then IOU_i will be the ratio of the area of $\hat{B}_i$ to the area of B_i. This provides a reasonable measure of the potential of the selected key-points in S_N to cover each ground truth bounding box. In operation there will be some variability depending on the selected clustering approach. Some points in S_{N_N} might also be included with clusters from S_{N_P} and bounding boxes for identified clusters might be increased by some margin before cropping.

Figure 17 shows the sorted IOU performance of each of the four image test sets using the $GFM\kappa$ method, along with the IOU performance of the R-CNN model with the same image test set. The plots for R-CNN are shown for two conditions: for R-CNN IOU the true IOU is shown, and for R-CNN ground truth overlap, only the overlap ratio for the ground truth bounding boxes is shown. The latter condition makes the comparison to the GFM methods equal in terms of measuring only the ground truth bounding box overlap for both R-CNN and GFM methods. For each bounding box B_i in the test set of 200 images, IOU_i is computed, and then these values are sorted to make the plot in Figure 17. The average number of bounding boxes per image varies from 1.6 for the ''cat'' class to 3.8 for the ''car'' class so the total number of bounding boxes per image set is between 320 and 760.

The R-CNN performance is generally better compared to S_{N_P} for a small N in the range of 100 to 200. However, IOU performance for GFM methods is non-decreasing as N increases, and when N is in the range of 400 to 4096 the S_{N_P}

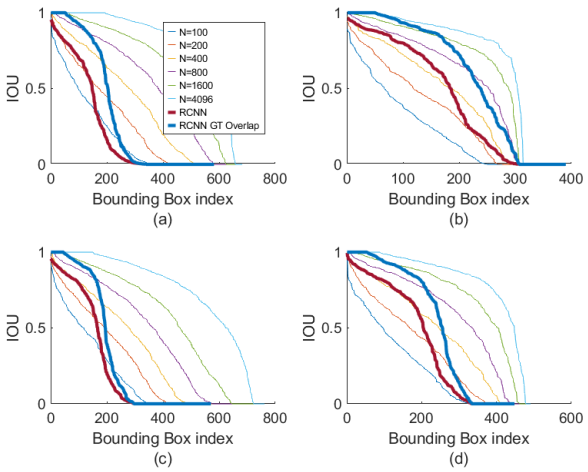

FIGURE 17. Sorted IOU performance compared to R-CNN IOU.

set is better than the R-CNN result. In addition, both R-CNN and GFM approaches have bounding box misses where IOU=0; the number of misses varies for different classes. The content-dependent variability for R-CNN is evident in Figure 17 in which the IOU values for ''cat'' and ''dog'' are considerably higher than the IOU values for ''boat'' and ''car''. While some ground truth bounding boxes are missed, the points in S_{N_N} may indicate the identification of potential object locations that are not part of the baseline test image data set.

One of the main methods for evaluating IOU ratios for bounding boxes in the VOC test environment is IOU@0.5, as described in [16]. IOU@0.5 assigns a value of 1 for a bounding box if the $IOU \geq 0.5$, and 0 otherwise. The average for the IOU@0.5 computed from Figure 17 as the percent of IOUs with values larger than 0.5 is summarized in Table 6. The average IOU@0.5 of the estimated bounding boxes is shown for a range of values of N. In addition the average IOU@0.5 was provided for the R-CNN model evaluated with the same test images and with the same IOU@0.5 threshold, for two R-CNN conditions as described above. In all class cases, the IOU@0.5 is low for N=100 and increases to a maximum for N=4096 since $||S_{N_P}||_0$ increases as N increases. At N=4096 the average IOU@0.5 ranges from 81.7 to 96.2, indicating that over 80% of the IOU measurements had an IOU score of 0.5 or better, for all classes. For N=100 and 200, the average IOU@0.5 for all classes is less than the results using R-CNN, which is consistent with the IOU comparison in Figure 17. However, for N greater than 800, the average IOU@0.5 for all classes using GFM methods is higher than the results using R-CNN. In general, R-CNN values fall between the results for N=200 and N=800.

Based on the results in Figure 17 and Table 6, clustering of selected points to form partitions for cropping should work well when N is above 400. The AP results in Table 4 show that although increasing N reduces AP, for GFM_{sum} and $GFM\kappa$

TABLE 6. Summary of IOU@0.5 performance.

Average IOU@0.5								
N	100	200	400	800	1600	4096	R-CNN	R-CNN Overlap
Boat	14.2	25.1	40.8	56.1	72.1	86.8	25.3	34.9
Cat	21.0	37.1	59.0	80.0	91.4	96.2	47.4	60.8
Car	15.1	25.4	36.5	47.8	60.4	81.7	29.2	34.0
Dog	19.5	33.9	49.8	67.9	81.9	90.9	45.5	56.9

the values are above 60% for N=1600 and only drop to a minimum of 58% for N=4096. Over the wide range of values of N between 400 and 4096 key-points are found which have a high probability of being inside an object based on AP and also cover a large area of the object bounding box. These partitions and cropped images would enable the VGG model to identify multiple objects in an image. Each cropped and rescaled image that encloses the isolated object would have a higher probability of being classified correctly by VGG than the original image because the probability of the cropped image including a more dominant object would be reduced. In addition the bounding box would provide a better estimate of the object location in the original image.

V. CONCLUSION

The objective of this work was to define and analyze a method that could identify sets of key-points in an image using a simple CNN architecture which had been trained for single object classification. The method developed did not rely on classification outputs, relevance back-propagation or training. It was based on the observation that early hidden layers have filters trained to respond to significant features across the full set of training data, and thus feature maps for a trained CNN model contain information that can be used to predict object key-points.

A good set of key-points would provide the input for well understood clustering methods, and bounding boxes based on clusters could provide partition definitions that would be used to crop the input image. The cropped images would be individually resized and classified by the CNN. Classifications with high confidence would be accepted, and the bounding box used for cropping would provide location information. This approach allows the identification of multiple objects in an image using a simple CNN architecture with a large number of classes. It does not require any specific retraining to predict object locations and does not require extensive iterative computation to establish the initial partitions.

Early feature maps were evaluated with emphasis on the feature maps prior to the first pooling layer in a CNN. In order to be effective, the sets of key-points should have a high probability of belonging to the bounding box of an object and should span enough of the space occupied by the object to provide reasonable initial bounding boxes. There is no need to identify all the points in the object. The selection strategy reflects the fact that the operation of max pooling

layers select the locally largest output before down-sampling and propagating to higher layers. The feature map before the first max pooling layer was analyzed in detail. Results using only this layer were better than using only the previous feature map layer or using only the feature map before the second pooling layer. Individual feature maps have a wide variation in output distributions depending on input image content, but for any specific image, it was demonstrated that in feature maps with high maximum values and high kurtosis values, the highest value outputs were more likely to be found within visually significant objects.

A Global Feature Map defined by combining the collection of feature maps in the selected layer using maximum or average values with and without kurtosis weighting provided the basis for selecting potential key-points to be the N highest values of the GFM. The GFM based on the average value over all feature maps was shown to outperform the GFM based on maximum value of all feature maps. The GFM based on kurtosis weighting gives priority to feature maps with distinct outliers, and it outperforms unweighted methods.

Evaluation of results used VOC, and training of the VGG model was done with ILSVRC trained for 1000 classes. Global Feature Maps were evaluated for a range of key-point set sizes from 100 to 4096. This range was shown to have good precision for up to 4096 selected points. A selection range of N greater than 4096 tended to select a higher percentage of points that were not key-points, which lowered the PPV, and did not significantly improve the IOU performance. To explore any variability due to types of objects in images, four classes from the VOC collection were selected as subsets for testing and analysis: ''boat'',''cat'', ''car'', and ''dog''.

IOU@50% was used to determine the percentage of area of each ground truth bounding box covered by the bounding box of selected points that were inside the bounding box. With a key-point set size between 800 and 4096, our approach compares favorably to the more complex and computationally costly R-CNN in the context of bounding box accuracy. In addition, our method often finds key-points for objects that are missed by R-CNN, and VGG may be able to identify these missed objects using larger class size. The bounding boxes used for cropping provide some location information which would not otherwise be available using VGG.

This key-point detection method has the potential to greatly improve the classification capability of simple CNNs by making it possible to identify multiple objects in a complex input image, with a modest computation cost, and also provide some localization information. This method was demonstrated with VGG, but since no unique characteristics of VGG were used in the development and analysis of the key-point detection method, it should be easily adapted for use on similar architectures which use max pooling. Different data sets were used for training the network and analyzing the key-point detection method, so it is expected that the method will be effective on a wide variety of data sources. In addition, since the method uses features in early layers it should extend without modification to identify objects for any object classification set developed with transfer learning.

REFERENCES

[1] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' in *Proc. ICLR*, 2015, pp. 1–14.
[2] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, ''Going deeper with convolutions,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[4] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, ''Selective search for object recognition,'' *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013.
[5] N. Dalal and B. Triggs, ''Histograms of oriented gradients for human detection,'' in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.
[6] D. G. Lowe, ''Distinctive image features from scale-invariant keypoints,'' *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ''You only look once: Unified, real-time object detection,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, ''SSD: Single shot MultiBox detector,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 1–17.
[9] S. Ren, K. He, and R. Girshick, ''Faster R-CNN: Towards real-time object detection with region proposal networks,'' in *Proc. CVPR*, 2016, pp. 1–9.
[10] Y. Liu, ''An improved faster R-CNN for object detection,'' in *Proc. 11th Int. Symp. Comput. Intell. Design (ISCID)*, Dec. 2018, vol. 2, pp. 119–123.
[11] T. Liang, H. Bao, W. Pan, X. Fan, and H. Li, ''DetectFormer: Category-assisted transformer for traffic scene object detection,'' *Sensors*, vol. 22, no. 13, p. 4833, 2022.
[12] T. Liang, H. Bao, W. Pan, and F. Pan, ''Traffic sign detection via improved sparse R-CNN for autonomous vehicles,'' *J. Adv. Transp.*, vol. 2022, Mar. 2022, Art. no. 3825532.
[13] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, and P. Luo, ''Sparse R-CNN: End-to-end object detection with learnable proposals,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14454–14463.
[14] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, ''Deep learning for generic object detection: A survey,'' *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020.
[15] M. Najibi, M. Rastegari, and L. S. Davis, ''G-CNN: An iterative grid based object detector,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2369–2377.
[16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, ''The PASCAL Visual Object Classes (VOC) challenge,'' *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[17] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, ''A survey of the recent architectures of deep convolutional neural networks,'' *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020.
[18] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, ''Object detectors emerge in deep scene CNNs,'' in *Proc. ICLR*, 2015, pp. 1–12.
[19] X. Ding, Q. Li, Y. Cheng, J. Wang, W. Bian, and B. Jie, ''Local keypoint-based Faster R-CNN,'' *Artif. Intell.*, vol. 50, pp. 3007–3022, Apr. 2020.
[20] X. Ding, Y. Luo, Y. Yi, B. Jie, T. Wang, and W. Bian, ''Orthogonal design for scale invariant feature transform optimization,'' *J. Electron. Imag.*, vol. 25, no. 5, Oct. 2016, Art. no. 053030.
[21] H. Bay, T. Tuytelaars, and L. Van Gool, ''SURF: Speeded Up Robust Features,'' in *Proc. 9th Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.
[22] N. J. S. Morch, U. Kjems, L. K. Hansen, C. Svarer, I. Law, B. Lautrup, S. Strother, and K. Rehm, ''Visualization of neural networks using saliency maps,'' in *Proc. Int. Conf. Neural Netw. (ICNN)*, 1995, pp. 2085–2090.
[23] K. Simonyan, A. Vedaldi, and A. Zisserman, ''Deep inside convolutional networks: Visualizing image classification models and saliency maps,'' in *Proc. ICLR Workshop*, 2014, pp. 1–8.
[24] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, ''On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation,'' *PLOS One*, vol. 10, no. 7, 2015, Art. no. e0130140.

[25] G. Montavon, S. Bach, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep Taylor decomposition," *Pattern Recognit.*, vol. 65, pp. 211–222, May 2017.
[26] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down neural attention by Excitation Backprop," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 1–21.
[27] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," Dept. CS OR, Univ. Montreal, Montreal, QC, Canada, Tech. Rep. 1341, 2009.
[28] M. Wojtas and K. Chen, "Feature importance ranking for deep learning," in *Proc. NeurIPS*, 2020, pp. 1–10.
[29] T. N. Mundhenk, B. Chen, and G. Fried-Land, "Efficient saliency maps for explainable AI," in *Proc. ICLR*, 2020, pp. 1–22.
[30] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.
[31] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, Feb. 2020.
[32] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *Proc. ICLR*, 2015, pp. 1–14.
[33] M. Kummerer, L. Theis, and M. Bethge, "Deep Gaze I: Boosting saliency prediction with feature maps trained on ImageNet," in *Proc. ICLR*, 2015, pp. 1–12.
[34] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, "Layer-wise relevance propagation: An overview," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning* (Lecture Notes in Computer Science). Saarbrücken, Germany: Max Planck Institute, 2019.
[35] J. K. Tsotsos, S. M. Culhane, W. Y. Kei Wai, Y. Lai, N. Davis, and F. Nuflo, "Modeling visual attention via selective tuning," *Artif. Intell.*, vol. 78, nos. 1–2, pp. 507–545, Oct. 1995.
[36] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001.
[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinv, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 5, no. 1, pp. 1929–1958, 2014.
[38] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt, "Feature selection via dependence maximization," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 1393–1434, Jan. 2012.
[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255. [Online]. Available: https://www.image-net.org/challenges/LSVRC/index.php
[40] S. Tammina, "Transfer learning using VGG-16 with deep convolutional neural network for classifying images," *Int. J. Sci. Res. Publications*, vol. 9, no. 10, p. p9420, Oct. 2019.
[41] M. Menikdiwela, C. Nguyen, H. Li, and M. Shaw, "CNN-based small object detection and visualization with feature activation mapping," in *Proc. Int. Conf. Image Vis. Comput. New Zealand (IVCNZ)*, Dec. 2017, pp. 1–5.
[42] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2018–2025.
[43] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.
[44] C. Aggrawal and C. Reddy, *Data Clustering Algorithms and Applications*. Boca Raton, FL, USA: CRC Press, 2014.
[45] J. A. Hartigan, *Clustering Algorithms*. Hoboken, NJ, USA: Wiley, 1975.

ALLEN RUSH (Member, IEEE) received the B.S. degree in electrical engineering from Lehigh University, Bethlehem, PA, USA, and the M.S. degree in computer engineering from Santa Clara University, Santa Clara, CA, USA. He has been holding several engineering positions with technology companies, with a focus on computer design, since 1974. He was a Senior Fellow with AMD, responsible for AI acceleration architecture and strategies for GPU and CPU products.

SALLY WOOD (Life Fellow, IEEE) received the B.S.E.E. degree from the Columbia School of Engineering and the M.S. and Ph.D. degrees in electrical engineering from Stanford University. Since 1985, she has been a Faculty Member with the Department of Electrical and Computer Engineering, Santa Clara University, where she is currently a Professor. Before joining as a Faculty Member with Santa Clara University, she was involved in research and development in medical imaging, neural systems modeling, and development of automatic reading systems with Contour Medical Systems, the Palo Alto Veterans Administration Rehabilitation Engineering Research and Development Center, and Telesensory Systems. She was the Program Director of the Directorate for Engineering, NSF, from 2008 to 2010. Her research interests include signal and image processing, computer vision, and computational imaging. She has been a member of the Board of Governors of the IEEE SPS and EMBS.