

This CVPR workshop paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version; the final published version of the proceedings is available on IEEE Xplore.

Improving Shape Awareness and Interpretability in Deep Networks Using Geometric Moments

Rajhans Singh Ankita Shukla Pavan Turaga Geometric Media Lab, Arizona State University

{rsingh70, ashukl20, pavan.turaga}@asu.edu



Figure 1. Feature visualization of different models on ImageNet. For the standard ResNet model, we use GradCAM for visualization. We also compare our visualization with the Vision Transformer [10] (ViT-B-16) attention map. Note that our DGM model produces a very sharp object shape.

Abstract

Deep networks for image classification often rely more on texture information than object shape. While efforts have been made to make deep-models shape-aware, it is often difficult to make such models simple, interpretable, or rooted in known mathematical definitions of shape. This paper presents a deep-learning model inspired by geometric moments, a classically well understood approach to measure shape-related properties. The proposed method consists of a trainable network for generating coordinate bases and affine parameters for making the features geometrically invariant yet in a task-specific manner. The proposed model improves the final feature's interpretation. We demonstrate the effectiveness of our method on standard image classification datasets. The proposed model achieves higher classification performance compared to the baseline and standard ResNet models while substantially improving interpretability.

1. Introduction

Advances in deep learning have resulted in state-of-theart performance for a wide variety of computer vision tasks. The large quantity of training data and high computation resources have made convolutional neural networks (CNN) into a common backbone model for many tasks; including image classification [21, 26, 32, 43], object detection [18, 20, 36], segmentation [2, 3, 37], unsupervised learning [5, 47], and generative modeling [19, 30, 42].

A CNN consists of multiple spatially compact filters which convolve over an input image, followed typically by normalizations [25] and nonlinearities. The convolutional kernel's small spatial extent and weight sharing properties make them efficient and translation equivariant. However, this also implies that the kernel's receptive field is limited due to its small spatial extent. The local nature of the convolution kernels prevents them from capturing the global context of the image. The long-range dependency, i.e., a larger receptive field, is achieved through stacking multiple CNN layers and reducing the spatial dimension by pooling operations.

However, it has been observed that the features from this kind of architecture tend to be more receptive toward texture than the shape of the object. For example, [17] tackles this problem using a better shape-biased dataset like Stylized-ImageNet. As opposed to this, incorporating the shape bias more directly without changing training sets is a natural choice. As we know, convolutional operations intrinsically represent frequency selective operations, while the shape is related to geometric concepts rather than specific frequency bands. Therefore, different types of operations that are more directly shape-sensitive are needed to promote shape awareness.

In this work, we frame vision tasks like classification through the geometrical properties of the object's shape. Rigid and non-rigid aspects of shape can be described in terms of geometric moments. Geometric moments are a very specific type of weighted averages of image pixel intensities, where the weights are drawn from specific polynomial-type basis functions. This operation can be expressed as a projection of the image on the bases. While classic theoretical development around moments used specific choices of the bases functions, their application to difficult tasks like image classification has remained very limited. In this paper, we revisit moments as a learnable spatial operation, introducing modules motivated by the image-projection analogy. However, we leave the bases to be learned end-to-end in a task-specific manner.

Geometric moments have a long history in the vision community for a wide variety of applications ranging from invariant pattern recognition [1, 14, 24, 28], segmentation [16, 35] and 3D shape recognition [11, 39, 46]. We propose a deeplearning based architecture to extract invariant image moments for classification tasks. Our architecture consists of two streams of convolutional networks; one extracts features corresponding to the object, i.e., removes the background from the image, and the second network learns the bases from a 2D coordinate grid. Geometric moments are computed by projecting features of the image to the learned bases. In order to learn task-specific invariant moments to deformations, size, and location, we learn a simple transformation of the coordinate grid and compute the geometric moments at multiple levels. The geometric moment captures long-range dependency without using any pooling layer or reducing the spatial dimension. The computation cost of the geometric moment is linear in the spatial dimension.

In particular, the proposed Deep Geometric Moment (DGM) architecture provides four key benefits compared to existing models.

 First, the model generates discriminative features for the classification task by accounting for shape information through the proposed deep geometric moments.

- Second, our model outperforms existing ResNet models on standard datasets without using any pooling layer or reducing the spatial dimension.
- Third, it provides easy access to interpretable features at any level by simple re-projection of moments.
- Finally, compared to existing models, the DGM model only requires finetuning the coordinate basis pipeline without retraining all the model parameters.

Our goal is not to outperform all the latest developments in vision, but to show that our proposed model can perform comparably to standard models when trained from scratch, produces interpretable results, and is easier to finetune.

2. Geometric moment

A *moment* for a given two-dimensional piece-wise continuous function f(x, y) is defined as:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy, \qquad (1)$$

where, (x, y) is the 2D coordinate and (p + q) is the order of the moment. By uniqueness theorem [24], if f(x, y) is a piece-wise bounded continuous function (i.e. it is non-zero only on a compact part of the xy plane), then the moment sequence m_{pq} is uniquely defined for all orders (p + q) by f(x, y). Conversely, f(x, y) is uniquely determined by the sequence m_{pq} .

Equivalently, moments can also be seen as a 'projection' of the 2D function on certain bases of the form $x^p y^q$. Instead of using the bases function of type $x^p y^q$, one can instead also use orthogonal functions like Legendre or Zernike polynomials [44] for better reconstruction.

Image moments are well-known invariant shape descriptors with a long history of use in the computer vision literature to capture the geometrical properties of an image. For example, m_{00} (0th order) represents average pixel intensity, m_{10} (1st order) and m_{01} (1st order) represent xy centroid coordinate and the combination of 1st and 2nd order can be used to compute orientation.

We need *discriminative moments*, which are also invariant to certain image transformations like rotation, translation, and scale for the image classification task. An early work by Hu [24] introduced a way to find invariant moments for images. The Hu moments consist of seven moments, mostly a combination of lower-order moments invariant under scaling, translation, and rotation. While these basic sets of seven Hu moments are provably invariant to rotation, translation, and scale, their use has been limited since their discriminative power is not very high. Developing invariant moments for the Legendre and Zernike polynomials for any arbitrary order is also possible [7, 13, 28, 29, 48, 51–53]. However, they also have not significantly impacted contemporary image classification tasks. In this paper, we seek to advance a new approach for defining spatial operations for image classification networks, whose structure is motivated by classic moment computation but whose basis functions are left to be learned end-to-end by a deep learning network in a task-specific way. This implies that we are not seeking to replicate any of the classical moments in an exact sense but to find ways to fuse momentlike computations and let networks learn the suitable basis functions for a given task. This approach is described in the next section.

Geometric moments and deep networks: There has been prior work in integrating geometric moments with deep networks, as specifically applied to 3D shape classification, from point-cloud data. For example, geodesic momentbased features from an auto-encoder were used to classify 3D shapes [34]. On the other hand, CNNs were used as a polynomial function to learn bases, and the needed affine transformation parameters for 3D point cloud data based shape classification [27]. This line of work was extended in [33] which uses graph CNN to capture local features of the 3D object. More recently, [45] and [49] replaced the conventional global average pooling in CNN models with invariant Zernike moment-based pooling for image classification task. Contrary to these methods, our approach learns bases as well as the affine parameters. Note that our work is different from these approaches because a) we are interested in natural image classification where moment computation is challenging compared to 3D shape classification, due to intensity variation, background variation, occlusions, etc, b) architecturally our approach is more involved compared to processing 3D object that are specified directly in terms of coordinate locations.

Spatial transformer network (STN) [26] predicts the affine transformation parameters for classifying images that help in maximizing object detection accuracy and transformation of the feature grid accordingly. The spatial transformation of the feature grid acts as an attention module and brings invariance to rotation, translation, scale, and different warping of the image data. [9] presents a spatially deformable convolution and pooling kernel to bring invariance under spatial transform. Recently, exotic techniques such as rotationally invariant convolution and regularization loss have emerged to incorporate invariance towards spatial transform the CNN feature grid and instead a 2D coordinate grid that is much simpler in terms of computation and implementation.

3. Deep geometric moments

CNNs are extremely good at capturing local context and texture information to discriminate images, even in a complex classification task, without an explicit 'shape' related operation. On the contrary, geometric moments can capture the shape information exceptionally well and provide discriminative cues in classifying images; however, their discrimination power is quite limited and generally requires a salient object over a homogeneous background. We advance a new type of architecture that blends the strengths of both approaches. We propose a deep geometric moment (DGM) model that uses geometric moments along with CNNs for classification by providing both shape and texture access. The geometric moment for a discrete 2D function is given by the discrete version of Eq. 1 :

$$m_{pq} = \sum_{x} \sum_{y} x^p y^q I(x, y) \tag{2}$$

In the traditional usage of moments in vision, the number and order of moments is an experimental design choice. Choosing the right number and order of moments depends on the underlying tasks; large moments are useful for image reconstruction, whereas, for image classification, higher-order moments are affected by noise and hence not very useful. Thus, selecting the correct moment orders is essential. In our method, we specify the required number of moments (in terms of feature dimension), but the exact basis functions and orders are learned by the networks end-to-end for specific tasks. Hence we will drop the subscript notation pq and use the superscript notation c indicating the feature channel number for moment m.

In our model defined by Eq. 3, we use CNNs to extract relevant object features from the given images and project them onto the learned coordinate bases *per channel*:

$$m^{c} = \frac{1}{N \times N} \sum_{x} \sum_{y} g^{c}(x, y) f^{c}(x, y), \qquad (3)$$

where, $N \times N$ is the dimension of the image, $g^c(x, y)$ is a learnable 2D polynomial function, and $f^c(x, y)$ is image feature at coordinate location (x, y), and c refers to the channel dimension of the feature, given by the CNN from the image-feature stream (see top-stream in Fig. 2). Next, to account for varying locations, sizes, poses, and deformation, we allow our network to learn affine parameters to appropriately deform the 2D coordinate grid during moment computation.

In summary, the proposed model consists of three components: 1) *Coordinate base computation*: uses a 2D coordinate grid as input and generates the bases, 2) *Image feature computation*: obtains image features through ResNet blocks, and 3) *Affine transform estimation*: to transform the 2D coordinate grid to enable invariance learning. An overview of the DGM model is shown in Fig. 2. The architecture consists of *Level-1* and *Level-2* blocks, where *Level-1* is fixed, whereas *Level-2* can be replicated multiple times to create deeper networks.

Coordinate base computation: For computing bases, expressed as g(x, y) in Eq. 3, a 2D coordinate grid is used



Figure 2. An overview of proposed Deep Geometric Moment (DGM) framework for image classification task. The model consists of two blocks: *Level-1* and *Level-2* that consists of two pipelines : 1) CNN based image feature extraction and 2) coordinate bases computation. The *Level-2* block can be repeated number of times for computing moments, similar to depth concept in deep networks.

as an input. The 2D coordinate grid is represented by $2 \times N \times N$, where $N \times N$ is the dimension of the input image. Each entry in the coordinate grid indicates the normalized 2D pixel locations $(x, y \in [0, 1]^2)$. g(x, y) in Eq. 3 is defined by a neural network that consists of two layers of 1×1 convolution layer followed by a batch-normalization and ReLU layer. This definition of g(x, y) processes each location of the coordinate grid independently. We use only two convolution layers in our experiments, but one can use more layers to learn more complex or higher-order moments. The output bases are of dimension $C \times N \times N$ where C is the number of moments/channels.

Image feature computation: Referring back to the term f(x, y) in Eq. 3, also shown as a ResNet block in Fig. 2, takes the image of dimension $3 \times N \times N$ as input, and outputs a feature of dimension $C \times N \times N$. f(x, y) is implemented as a conventional ResNet block [21] with 3×3 filter kernel. Note that the geometric moments are insufficient for capturing local features in the image. In contrast, the CNNs with a kernel size of 3×3 or greater are very efficient in capturing local properties. Therefore, we use a kernel of size 3×3 in ResNet blocks, which is a common choice in state-of-the-art ResNet-based models. These image features from the ResNet block are then projected on the bases by performing element-wise multiplication, and moments are obtained by using global pooling on the projected features. Note that unlike the conventional definition of geometric moment Eq. 1 where the same 2D image is projected to each basis, in our method, different feature maps are projected to each basis. The projected feature map on the bases highlights the important region in the feature map.

Affine transformation estimation: We first use the canonical coordinate grid to compute the moments and

predict the affine parameters using these moments. The prediction network consists of two fully-connected layers with a non-linear activation. The prediction network takes the canonical moments $(1 \times C)$ as input and outputs each feature channel's affine parameters $(C \times 6)$. Then, the 2D coordinate grid C_G is transformed according to: $C'_G = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \times C_G + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$, where, the 2×2 matrix is the predicted affine parameters, t_x and t_y are the predicted translation parameters and C'_G is the transformed 2D coordinate grid. We then use these transformed coordinate grids to generate new bases and compute new moments for each channel. Arguably, affine parameters are also limited in providing needed invariance and robustness, but this choice is efficient and leads to good performance.

3.1. DGM classification model

The proposed DGM network for the image classification task is shown in Fig. 2. It uses the computed coordinate bases, convolution network-based features, and affine transformation estimation and is trained end-to-end. The functionality of the model comprises of: 1) an image feature pipeline that transforms an image to features through ResNet blocks, and 2) a geometric moment pipeline that generates the bases and computes the affine parameters and moments. The proposed model does not use any pooling layer or reduce the spatial dimensions across the networks. This preserves the shape of the object, as opposed to pooling or reducing the spatial dimension that distorts the final reconstructed shape, limiting its interpretability. For simplicity, we also use the same number of feature channels in each ResNet layer.

As shown in Fig. 2, *Level-1* uses the canonical coordinate grid to generate bases and the ResNet block to generate

features from the image. We then project this feature on the bases to compute the moments. The projected feature acts as an attention map and is added to the original feature. This feature map and geometric moments are then passed to *Level-2*.

The *Level-2* contains a ResNet block to process the features further. This level also predicts the affine parameters based on the moments from the previous level and transforms the coordinate grid to regenerate the bases. Fig. 2 shows only two levels, but one can repeat the *Level-2* block multiple times for added depth. The moments from the final level are used as input to the fully-connected layer to generate class probabilities for the classification task.

Feature Visualization: To visualize the shape awareness brought by the DGM approach, we describe a particular to visualize the learned features that highlight the object's shape. By the uniqueness theorem [24], moments can be used to reconstruct the original input, provided the bases are complete. In our case, our learned bases are under-complete. Using the moments as combination weights on the projected features given by:

$$V = \sum_{c} m^{c} (G^{c} \otimes F^{c}), \tag{4}$$

where, m^c is the moment, G^c is the basis, F^c is the image feature for channel c, and \otimes is element-wise multiplication, we get a visualization of shape-related information in the features.

4. Experimental results

We evaluate the proposed method for image classification on standard datasets: CIFAR-10, CIFAR-100 [31], and ILSVRC-2012 ImageNet [38] to validate the effectiveness of our model. The performance of our model is compared to a baseline model (ResNet model without pooling layers) and standard ResNet models [21] across classification accuracy (in %) and the number of parameters (in Million M). Along with the classification performance, we also compare the feature reconstruction qualitatively and semantic segmentation. We train all models under the same training hyperparameters. For CIFAR datasets, we train models up to 150 epochs with a batch size of 128; for ImageNet, we train for 100 epochs with a batch size of 256. We use SGD optimizer with momentum = 0.9, weight_decay = $5e^{-4}$, and cosine learning rate decay with an initial learning rate of 0.1.

4.1. How many levels do we need?

In this experiment, the ResNet block in the image feature pipeline consists of only 1×1 filter kernels, except the first Conv layer. There is no interaction between neighboring pixels in this setting; the only interaction is through geometric

Table 1. Performance comparison of DGM model with increasing levels on CIFAR datasets

| Model | Params (M) | CIFAR 10 (%) | CIFAR 100 (%) |
|--------------------------|---------------|-----------------|------------------|
| DGM Level-1 | 0.44 | 84.79 | 59.07 |
| DGM Level-2 | 1.37 | 88.77 | 68.07 |
| DGM Level-3 | 2.30 | 90.09 | 69.72 |
| DGM Level-4 | 3.23 | 90.28 | 70.56 |
| DGM Level-4 (w/o affine) | 2.03 | 88.47 | 66.6 |

moments and affine transformation of the coordinate grid. Each level in our DGM model consists of a ResNet block with two ResNet layers, a coordinate bases generator defined with two convolutional layers, and two fully connected layers for affine parameters prediction. This setting helps us understand the overall contributions of the coordinate bases and the affine transformation.

Table 1 reports the DGM model's classification performance and the number of parameters as we increase the number of levels in the model. In Level-1, we use the canonical coordinate grid to generate the bases. The results show that DGM Level-2 performs significantly better than DGM Level-1 on both the CIFAR-10 and CIFAR-100 datasets. This performance improvement reflects the effectiveness of transforming the coordinate grid to regenerate better bases. We also see that the performance difference between the Level-3 and Level-4 model is minimal. In DGM Level-4 w/o affine transform (last row), we do not transform the coordinate grid; hence the bases in this case, remain the same for every image. Without affine transformation, the model's performance drops, indicating the effectiveness of using affine transformation. Also, increasing the levels beyond 4 does not significantly improve the classification performance but is accompanied by a large increase in the number of parameters and computation; hence, we use only Level-4 in all the experiments.

4.2. Comparison with baseline ResNet model

Table 2. Performance comparison of proposed DGM model and baseline ResNet model (without pooling layers) on CIFAR datasets

| Model | Params | CIFAR | CIFAR |
|-------------------------|--------|--------------|--------------|
| | (M) | 10 (%) | 100 (%) |
| ResNet-18 (w/o pooling) | 9.62 | 94.78 | 76.93 |
| DGM ResNet-18 | 11.61 | 95.51 | 80.60 |
| ResNet-34 (w/o pooling) | 18.94 | 95.46 | 78.42 |
| DGM ResNet-34 | 21.06 | 96.27 | 82.13 |

The baseline model in this experiment is constructed in the same manner as our DGM model but without projection onto the coordinate bases. The baseline models are similar to standard ResNet models without pooling layers or reducing the spatial dimension. Without the reduction

Table 3. Performance comparison of proposed DGM model with baseline ResNet model (without pooling layers) on ImageNet dataset

| Model | Params (M) | Accuracy (%) |
|-------------------------|---------------|-----------------|
| ResNet-18 (w/o pooling) | 9.89 | 68.42 |
| DGM ResNet-18 | 11.88 | 72.36 |
| ResNet-34 (w/o pooling) | 19.20 | 73.34 |
| DGM ResNet-34 | 21.32 | 75.63 |

in the feature's spatial dimension, the receptive field of the filter kernels reduces and hence weakens the long-range dependency captured by the model. This experiment helps us establish that the moments effectively capture long-range dependency without reducing the spatial dimension of the features. We use global pooling on the features from the last ResNet layer of the baseline model to get the final feature vector for classification.

In Table 2, the baseline ResNet-18 and ResNet-34 models are based on the standard ResNet models with a constant number of feature channels (256) in every layer and without any pooling layers or reducing the spatial dimensions. DGM ResNet-18 and DGM ResNet-34 are also of 4 levels, with coordinate grid size of 32×32 . Table 2 shows that the proposed DGM models perform much better than the baselines on both CIFAR datasets. The performance improvement validates the effectiveness of using coordinate bases pipeline.

For comparison on the ImageNet dataset, we use ResNet-18 and ResNet-34 type architectures on a grid size of 32×32 . We divide the 256×256 image into 8×8 patches and use a linear embedding layer similar to Vision Transformer (ViT) [10] to reduce the spatial dimension to 32×32 , followed by the DGM model. For the embedding layer, we use a convolution layer with 8×8 kernel and stride of 8. Both baseline and DGM models consist of 3×3 filter kernels and 256 feature channels and are trained under the same hyperparameters. Table 3 shows that the proposed DGM model provides an improvement of $\sim 4\%$ (in the case of ResNet-18) over the baseline model with the same feature extraction pipeline. The performance of baseline models is less than the standard ResNet models. This performance reduction in the baseline models is mainly due to no pooling layers or reduction in features' spatial dimension, which results in a low receptive field. Our DGM model's performance is comparable to the standard ResNet model but without a spatial dimension reduction, showing that moments effectively capture the required long-range dependencies.

4.3. Comparison with standard ResNet model

Table 4 and 5 compare the proposed DGM model with conventional ResNet models [21]. The number of feature channels in the standard ResNet model is

Table 4. Performance improvements of DGM models over standard ResNet model (with pooling layers) on CIFAR datasets

| Model | Params (M) | CIFAR- 10(%) | CIFAR- 100(%) |
|----------------|---------------|-----------------|------------------|
| ResNet-18 | 11.17 | 95.37 | 77.35 |
| DGM ResNet-18 | 11.61 | 95.51 | 80.60 |
| ResNet-34 | 21.33 | 95.58 | 78.83 |
| DGM ResNet-34 | 21.06 | 96.27 | 82.13 |
| MobileNet [23] | 3.93 | 93.62 | 73.53 |
| DGM MobileNet | 4.50 | 96.33 | 82.19 |

Table 5. Performance comparison of DGM model with standard ResNet model (with pooling layers) on ImageNet dataset

| Model | Params (M) | Accuracy (%) |
|----------------|---------------|-----------------|
| ResNet-18 | 11.69 | 71.23 |
| DGM ResNet-18 | 11.88 | 72.36 |
| ResNet-34 | 21.80 | 74.58 |
| DGM ResNet-34 | 21.32 | 75.63 |
| ResNet-50 | 25.56 | 76.92 |
| DGM ResNet-50 | 23.51 | 77.06 |
| MobileNet [23] | 4.20 | 70.66 |
| DGM MobileNet | 4.76 | 72.69 |

(64, 128, 256, 512); whereas, in our DGM model, the number of feature channels is 256 in ResNet-18 and Resnet-34 and 512 in ResNet-50 across all levels. The naming of our DGM model is based on the number of layers used in the image feature pipeline (similar to the standard ResNet model naming convention). The standard ResNet models use pooling layers to reduce the spatial dimensions up to 8×8 in the final layer, whereas, in our DGM model, the final feature's spatial dimension is 32×32 .

Table 4 shows that the proposed DGM models perform better than the standard ResNet models on both CIFAR-10 and CIFAR-100 datasets. The DGM model improves the accuracy of ~ 1% on CIFAR-10 and ~ 3% on CIFAR-100. Table 5 compares the DGM model with the standard ResNet model on the ImageNet dataset. We observe that our DGM model is better than the standard ResNet model while using a similar number of parameters but without using any pooling layers.

The computation cost in our DGM model is much higher than conventional ResNet, which is attributable to the fact that we do not reduce the spatial dimension of the image features. However, one can reduce the computation cost in the image pipeline using the channel-wise convolution [23, 40]. In Table 4 and 5, DGM MobileNet has very less number of parameters, but performs comparable to DGM model with conventional ResNet layers. Details about the computation cost of our model are provided in the supplementary.



Figure 3. Visualization at different levels for DGM ResNet-34 model on the ImageNet dataset. We note that at higher levels our model is able to separate the background information from the object's shape compared to initial levels.

4.4. Feature visualization

A major advantage of keeping a higher spatial dimension of features is interpretability at different levels. The feature vectors can be easily visualized by a reconstruction step as given by Eq. 4. Fig. 1 compares Level-4 visualization as a heatmap for a few randomly selected images from the ImageNet dataset for our DGM model against baseline ResNet and standard ResNet models. The visualization for the baseline model is just a weighted sum of the final Conv layer activation based on the global feature. For the standard ResNet model, we use GradCAM [41]. We also compare our visualization with the current attention-based Vision Transformer [10] (ViT-B-16) model, which is pre-trained on the ImageNet-21K and finetuned on ImageNet-1K datasets. As shown in Fig. 1, the GradCAM visualizations of the standard ResNet-18 model generate a blob-like shape around the critical region in the image, with no discernible object shape. While it gets better for the baseline ResNet-18 model, the heatmaps are still diffuse, and the shapes are not very distinct. However, with DGM model, object shapes are crisp, with improved classification accuracies. Also, our heat map is much sharper than the vision transformer attention map (Vit-B-16).

Additionally, in the DGM model, we can visualize features at different levels providing much better-debugging capability, as shown in Fig. 3. At initial levels, the heatmap is noisy, and the model is not able to able to separate the object from the background clearly as compared to higher levels. Additional qualitative results for analysis are provided in Supplementary.

4.5. Finetuning

For DGM finetuning, we only need to retrain the coordinate bases pipeline, and the final classifier layer, while

Table 6. Classification performance on fine-tuning of pre-trained ImageNet DGM model on CIFAR datasets

| Model | Params (M) | CIFAR 10(%) | CIFAR 100(%) |
|---------------|---------------|----------------|-----------------|
| DGM ResNet-18 | 11.69 | 93.79 | 75.83 |
| DGM ResNet-34 | 21.32 | 94.01 | 77.51 |
| DGM MobileNet | 4.76 | 93.87 | 75.92 |

| Table 7. Mean Corruption Error | (mCE) comparison of DGM with |
|--------------------------------|------------------------------|
| standard ResNet model on Image | eNet-c dataset |

| Model | Params | Clean↑ | mCE↓ |
|---------------|--------|----------|--------------|
| | (M) | Acc. (%) | (%) |
| ResNet-50 | 25.56 | 76.92 | 74.97 |
| DGM ResNet-50 | 23.51 | 77.06 | 71.74 |

freezing the image feature exaction pipeline. The coordinate bases pipeline contains significantly fewer parameters and requires less computation. We finetune our ImageNet pretrained DGM model for only 30 epochs on CIFAR-10 and CIFAR-100 datasets. The network is finetuned using a SGD optimizer with a cosine decay learning rate and an initial learning rate of 0.01. Table 6 shows the accuracy of the finetuned model on the CIFAR-10 and CIFAR-100 datasets. The performance of the finetuned model drops as compared to DGM trained from scratch, but it performs equally well to the standard ResNet model trained from scratch.

4.6. Performance under color distortion

We evaluate the effect of color distortions on DGM performance by testing it on ImageNet-C [22]. The ImageNet pretrained DGM model used for this experiment does not use any color augmentation during training, making it a sufficiently challenging task. The DGM ResNet-50 and Grad-CAM ResNet-50 visualization for two distortions is shown



Figure 4. Visualization from DGM and standard ResNet model under two different color distortion (Blur (1^{st} row) and Fog (4^{th} row) from Imagenet-C). Our model $(3^{rd} \text{ and } 6^{th} \text{ row})$, is able to produce consistent shape across different distortion compared to standard ResNet $(2^{nd} \text{ and } 5^{th} \text{ row})$.

in Fig. 4. The figure shows that our model captures the object shape very well under different challenging distortions like fog and blur. The GradCAM heatmap for the ResNet-50 is not very consistent across distortions as compared to our DGM model. For quantitative performance, we use the mean Corruption Error (mCE) metric (lower is better) [22]. We choose our DGM model such that it performs comparably to the standard ResNet-50 model on clean images but with fewer parameters. Table 7 shows DGM model provides an improvement of 3.2 points on the corrupted images.

4.7. Semantic image segmentation

The DGM model is evaluated on the PASCAL VOC 2012 [12] and Cityscapes [8] semantic segmentation benchmark datasets. We use a Level-5 DGM model with ResNet-50 as the image-feature pipeline pretrained on the ImageNet dataset. The number of parameters in the Level-5 DGM ResNet-50 is almost identical to the standard ResNet-50 model (~ 25 M). We use the same training hyperparameters as in the DeepLabv3+ model [4]. We test the effectiveness of our model with two different segmentation heads; first, with just two 1×1 Conv layers as segmentation head that takes the final 2D features rescaled by factor of 4, and second, DeepLabv3+ segmentation head, which consists of atrous convolution with different rates to capture long range dependencies. In Table 8, we observe that our model performs comparable to the standard DeepLabv3+ model, even with a very simple segmentation head on both datasets. This shows that geometric moments are effective in capturing long range dependencies. Our model shows improvements of 1.5%points on Pascal VOC and 0.7% points on Cityscapes Val sets compared to the standard ResNet model.

Table 8. Semantic segmentation perfomance on PASCAL VOC 2012 [12] and Cityscapes [8] val set in terms of mean intersection over union (mIoU)

| Backbone | Segmentation head | PASCAL VOC (mIoU) | Cityscapes (mIoU) |
|---------------|-------------------|----------------------|----------------------|
| ResNet-50 | 1×1 Conv | 68.59 | 71.92 |
| DGM-ResNet-50 | 1×1 Conv | 78.43 | 74.77 |
| ResNet-50 | DeepLabv3+ | 78.36 | 75.34 |
| DGM-ResNet-50 | DeepLabv3+ | 79.89 | 76.03 |

5. Conclusion

In this work, we propose a geometric moment-based deep learning model that explicitly captures shape-related information in an end-to-end learnable fashion. The DGM model improves the interpretability of features while also learning discriminative features. The quantitative and qualitative results on standard image classification and segmentation datasets show that our method performs better than the corresponding baseline and standard ResNet models. In addition, the DGM model provides easy interpretability at different levels while also providing ease of finetuning on a given dataset. Further, our model captures the object's shape even under extreme affine and color aberrations while performing better than existing approaches. We believe that DGM can improve the performance of other vision tasks, such as object detection and generation, and can be generalized to other modalities like video, RGBD, and volumetric data.

Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00112290073. Approved for public release; distribution is unlimited.

References

- Naif Alajlan, Mohamed S Kamel, and George H Freeman. Geometry-based image retrieval in binary image databases. *IEEE transactions on pattern analysis and machine intelli*gence, 30(6):1003–1013, 2008. 2
- [2] Liang-Chieh Chen, Maxwell D Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jonathon Shlens. Searching for efficient multi-scale architectures for dense image prediction. In *NeurIPS*, 2018.
 1
- [3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1
- [4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision* (ECCV), pages 801–818, 2018. 8
- [5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2180–2188, 2016. 1
- [6] Gong Cheng, Peicheng Zhou, and Junwei Han. Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7405–7415, 2016. 3
- [7] Chee-Way Chong, Paramesran Raveendran, and Ramakrishnan Mukundan. Translation and scale invariants of legendre moments. *Pattern recognition*, 37(1):119–129, 2004. 2
- [8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 8
- [9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference* on computer vision, pages 764–773, 2017. 3
- [10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference* on Learning Representations, 2020. 1, 6, 7
- [11] Asi Elad and Ron Kimmel. On bending invariant signatures for surfaces. *IEEE Transactions on pattern analysis and machine intelligence*, 25(10):1285–1295, 2003. 2
- [12] Mark Everingham, SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 8

- [13] Jan Flusser, Jirí Boldys, and Barbara Zitová. Moment forms invariant to rotation and blur in arbitrary number of dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):234–246, 2003. 2
- [14] Jan Flusser and Tomas Suk. Pattern recognition by affine moment invariants. *Pattern recognition*, 26(1):167–174, 1993.
 2
- [15] Patrick Follmann and Tobias Bottger. A rotationally-invariant convolution module by feature map back-rotation. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 784–792. IEEE, 2018. 3
- [16] Alban Foulonneau, Pierre Charbonnier, and Fabrice Heitz. Affine-invariant geometric shape priors for region-based active contours. *IEEE transactions on pattern analysis and machine intelligence*, 28(8):1352–1357, 2006. 2
- [17] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2018. 2
- [18] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1
- [19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1
- [20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017. 1
- [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 1, 4, 5, 6
- [22] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018. 7, 8
- [23] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017. 6
- [24] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2):179–187, 1962. 2, 5
- [25] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 1
- [26] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025, 2015. 1, 3
- [27] Mor Joseph-Rivlin, Alon Zvirin, and Ron Kimmel. Momen(e) t: Flavor the moments in learning to classify shapes. In

Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pages 0–0, 2019. 3

- [28] Alireza Khotanzad and Yaw Hua Hong. Invariant image recognition by zernike moments. *IEEE Transactions on pattern analysis and machine intelligence*, 12(5):489–497, 1990.
 2
- [29] Hyung Shin Kim and Heung-Kyu Lee. Invariant image watermark using zernike moments. *IEEE transactions on Circuits* and Systems for Video Technology, 13(8):766–775, 2003. 2
- [30] Diederik P Kingma and Max Welling. Stochastic gradient vb and the variational auto-encoder. In Second International Conference on Learning Representations, ICLR, volume 19, page 121, 2014. 1
- [31] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009. 5
- [32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097– 1105, 2012. 1
- [33] Dilong Li, Xin Shen, Yongtao Yu, Haiyan Guan, Hanyun Wang, and Deren Li. Ggm-net: Graph geometric moments convolution neural network for point cloud shape classification. *IEEE Access*, 8:124989–124998, 2020. 3
- [34] Lorenzo Luciano and A Ben Hamza. Deep learning with geodesic moments for 3d shape classification. *Pattern Recognition Letters*, 105:182–190, 2018. 3
- [35] Anthony P. Reeves, Richard J Prokop, Susan E. Andrews, and Frank P. Kuhl. Three-dimensional shape analysis using moments and fourier descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):937–943, 1988. 2
- [36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 1
- [37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1
- [38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 5
- [39] Firooz A Sadjadi and Ernest L Hall. Three-dimensional moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):127–136, 1980. 2
- [40] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE* conference on computer vision and pattern recognition, pages 4510–4520, 2018. 6
- [41] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Gradcam: Visual explanations from deep networks via gradientbased localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 7

- [42] Rajhans Singh, Pavan Turaga, Suren Jayasuriya, Ravi Garg, and Martin Braun. Non-parametric priors for generative adversarial networks. In *International Conference on Machine Learning*, pages 5838–5847. PMLR, 2019. 1
- [43] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1–9, 2015. 1
- [44] Michael Reed Teague. Image analysis via the general theory of moments. *Josa*, 70(8):920–930, 1980. 2
- [45] Thomas Theodoridis, Kostas Loumponias, Nicholas Vretos, and Petros Daras. Zernike pooling: Generalizing average pooling using zernike moments. *IEEE Access*, 9:121128– 121136, 2021. 3
- [46] Mihran Tuceryan. Moment-based texture segmentation. Pattern recognition letters, 15(7):659–668, 1994. 2
- [47] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European Conference* on Computer Vision, pages 268–285. Springer, 2020. 1
- [48] Lizhi Wang and Glenn Healey. Using zernike moments for the illumination and geometry invariant classification of multispectral texture. *IEEE Transactions on Image Processing*, 7(2):196–203, 1998. 2
- [49] Jiasong Wu, Shijie Qiu, Youyong Kong, Yang Chen, Lotfi Senhadji, and Huazhong Shu. Momentsnet: a simple learningfree method for binary image recognition. In 2017 IEEE International Conference on Image Processing (ICIP), pages 2667–2671. IEEE, 2017. 3
- [50] Wenju Xu, Guanghui Wang, Alan Sullivan, and Ziming Zhang. Towards learning affine-invariant representations via data-efficient cnns. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 904–913, 2020. 3
- [51] Pew-Thian Yap and Raveendran Paramesran. An efficient method for the computation of legendre moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1996–2002, 2005. 2
- [52] Hui Zhang, Huazhong Shu, Gouenou Coatrieux, Jie Zhu, QM Jonathan Wu, Yue Zhang, Hongqing Zhu, and Limin Luo. Affine legendre moment invariants for image watermarking robust to geometric distortions. *IEEE Transactions on Image Processing*, 20(8):2189–2199, 2011. 2
- [53] Hui Zhang, Huazhong Shu, Guoniu N Han, Gouenou Coatrieux, Limin Luo, and Jean Louis Coatrieux. Blurred image recognition by legendre moment invariants. *IEEE Transactions on Image Processing*, 19(3):596–611, 2009. 2