A Simple Approach to Image Tilt Correction with Self-Attention MobileNet for Smartphones

Siddhant Garg[1]
siddhantgarg@umass.edu
Debi Prasanna Mohanty[2]
debi.m@samsung.com
Siva Prasad Thota[2]
siva.prasad@samsung.com
Sukumar Moharana[2]
msukumar@samsung.com

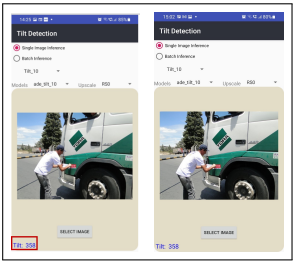
[1] University of Massachusetts Amherst, USA
[2] On-Device AI Samsung Research, Bengaluru, India

Abstract

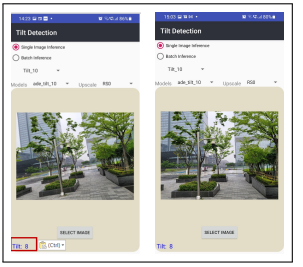
Main contributions of our work are two-fold. First, we present a Self-Attention MobileNet, called SA-MobileNet Network that can model long-range dependencies between the image features instead of processing the local region as done by standard convolutional kernels. SA-MobileNet contains self-attention modules integrated with the inverted bottleneck blocks of the MobileNetV3 model which results in modeling of both channel-wise attention and spatial attention of the image features and at the same time introduce a novel self-attention architecture for low-resource devices. Secondly, we propose a novel training pipeline for the task of image tilt detection. We treat this problem in a multi-label scenario where we predict multiple angles for a tilted input image in a narrow interval of range 1° or 2°, depending on the dataset used. With the combination of our novel approach and the architecture, we present state-of-the-art results on detecting the image tilt angle on mobile devices as compared to the MobileNetV3 [9] model. Finally, we establish that SA-MobileNet is more accurate than MobileNetV3 on SUN397 [30], NYU-V1 [21] and ADE20K [35] datasets by 6.42%, 10.51%, and 9.09% points respectively and faster by at least 4 milliseconds on Snapdragon 750 Octa core.

1 Introduction

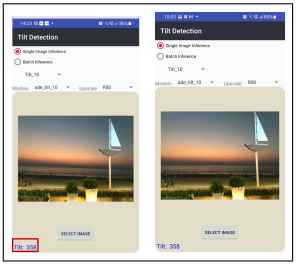
Smartphones have become the most convenient way to capture high-quality photos and videos. Mobile phone cameras have evolved over the past years with both hardware as well as software improvements with AI-enabled technologies that allow the users to take extremely high-resolution images. But many of us are not professional photographers and usually take images that are slightly skewed from the exact upright orientation. This reduces the aesthetic quality of images and people want that their holiday snapshots are of the highest



(a) Image is rotated 2° anti-clockwise to align vertical lines on the truck



(b) Image is rotated 8° clockwise to make it upright.



(c) Image is rotated 2° anti-clockwise to make the horizon horizontal.

Figure 1: Results on Image Tilt Correction in Real-Time. Proposed model is able to detect large as well as finer tilt angles. The predicted tilt angles are shown in a red box at bottom-left area of the images. Tilt angle value inside the box implies that image is tilted by that value in anti-clockwise direction from the upright orientation.

quality. Professional photographers use softwares like Lightroom or Photoshop to straighten the tilted vertical or horizontal lines.

We present an On-Device AI solution for automatic tilt angle detection of smartphone images and correct their orientations with a click to improve the overall picture quality. The proposed model is able to make inferences using mobile CPUs or GPUs with low latency values and at the same time respect the privacy of the user by removing the need to upload images to a server for processing. Currently, MobileNetV3 [9] networks are the most popular lightweight models for mobile devices for many computer vision tasks like image classification or object detection.

In this paper, we are proposing Spatial Self-Attention Modules that can learn long-range dependencies and global context within the input images. Furthermore, to enable on-device inference on resource-limited devices, we integrated these modules with the Inverted Bottleneck blocks [20] of MobileNetV3 to give us a novel neural network architecture for mobile devices called **Self-Attention MobileNet** or SA-MobileNet. The proposed network is able to learn the spatial information and overcomes the limitations of traditional convolutional kernels that only looks for different features in an image and not their relative positioning.

We are also proposing a simple yet effective training approach, described in Section 3.2, to handle the image tilt detection problem that scales to variety of image datasets containing natural, or indoor/outdoor images. The combination of the Self-Attention MobileNet and proposed training approach gives us state-of-the-art results for detecting image tilt for mobile devices in real-time.

Therefore, our two main contributions are:

- Self-Attention MobileNet for mobile/IoT devices for real-time inference.
- A novel approach to tackle fine-grained Image Tilt Detection problem.

2 Related Works

Image Tilt Angle Detection is a long-standing problem. Before the advent of deep learning, low-level image features were used to detect the upright image orientation like Ciocca *et al*. [3] who used LPB-based image features and logistic regression for this problem. But

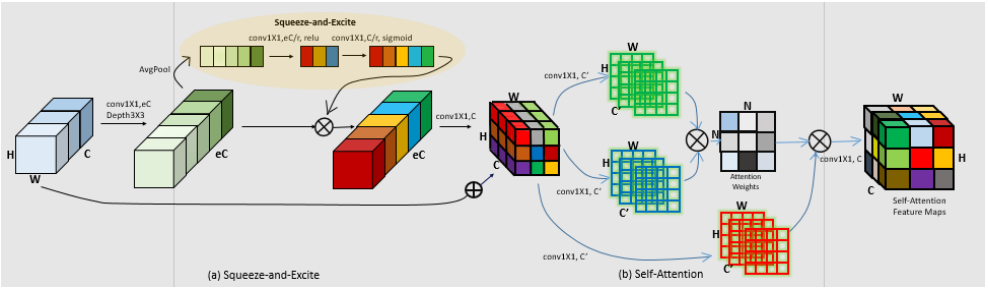


Figure 2: Complete Attention Module with Squeeze-and-Excite and Spatial Self-Attention. (a) e: expansion factor, (b) $C' = C/r$, r: reduction ratio ($r = 8$). Best viewed in color.

when the deep learning models for computer vision became popular, researchers started using high-level image features [8, 28, 33]. Fischer *et al*. [6] used AlexNet [11] to regress the exact orientation angle of tilted images but there angle error was high($\approx 20°$) for complete 360° range of image orientations. Applying CNNs for coarse-angle estimation and fuzzy logic for precise angle estimation on edge pixels [17, 18] was recently employed to take into account the ambiguity and uncertainty in image orientations. Horizon lines and vanishing points are also used as cues for optimal image tilt detection but these methods are generally limited to outdoor images with a clear horizon line [5, 28] whereas our work addresses natural images in diverse environments.

Digital camera parameters from accelerometer data are also used to rectify the image orientations. Do *et al*. [4] proposed *spatial rectifier* with ResNet-18 [7] backbone network for surface normal estimation of indoor images. G Olmschenk *et al*. [14] proposed an InceptionNet [24] style architecture for estimating the pitch and roll of the camera from a single 2D image. Xian *et al*. [29] used surface geometry to determine surface normals for estimating 2DoF [23] camera orientations. But all of the above methods use neural networks that need high computational resources and it is not possible to deploy them in mobile/IoT devices.

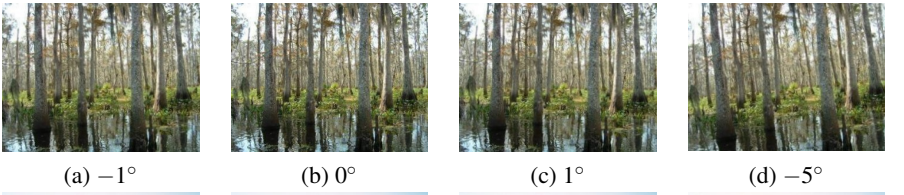
Self-Attention has also gained a lot of popularity in recent years. It has quickly become state-of-the-art baselines for many NLP tasks [2, 15, 26] and after that it has also became popular in many computer vision tasks like Image Captioning [31], Image Question Answering [32], and Object Detection [1]. Self-Attention modules can model long-term dependencies and overcome the limitation of convolutional kernels which operate in a local neighbourhood [34]. BAM [16], CBAM [27], and ULSAM [19] are light-weight attention modules for spatial attention but they use pooling operations that result in loss of information of the feature maps.

3 Method

We now present the architecture of our proposed Self-Attention MobileNet model and the novel training pipeline for Image Tilt Detection in detail in this section.

3.1 Self-Attention Convolutional Module

Squeeze-and-Excite [10]: The squeeze operation captures channel-wise dependencies in a feature map by using global average pooling operation along the channel dimension to ag-



(a) -1° (b) 0° (c) 1° (d) -5°

(e) -1° (f) 0° (g) 1° (h) 5°

Figure 3: Images (a), (b), (c) and (e), (f), (g) can be perceived as upright because of very small deviations from the upright orientation. But the Fig.(d) needs to be rotated 5° clockwise and Fig.(h) by 5° anti-clockwise to make them upright.

gregate information of each 2D feature map through a scalar value that results in a channel descriptor vector. During the excite operation, we learn channel-wise dependencies by passing the channel descriptor vector through a 2-layer neural network which outputs channel-wise attention weights. This attention vector is then multiplied by the input feature map to adaptively weight each channel and improve the representational power of the feature maps.

Spatial Self-Attention: Standard convolutional kernels process a local neighborhood of an image at a time because of the smaller receptive field sizes for the input feature maps. Although, the kernels with large receptive fields can cover more region but that will incur high computational costs because of the increased number of parameters and training time. Self-attention modules can complement convolutional layers by **learning global or long-range dependencies** between different image regions without much computational overhead. To learn spatial self-attention, we take the feature map vectors along the channel dimension and calculate their key, query, and value representations in a low-dimensional subspace. The dot product of all key vectors with every other query vector with the softmax function gives us the attention weights between all the regions of the image. More specifically, after squeeze and excite operation, let the feature map be $\mathbf{F} \in \mathbb{R}^{\mathbf{H}\times\mathbf{W}\times\mathbf{C}}$ and after expanding it along spatial dimensions we get the matrix $\mathbf{F} \in \mathbb{R}^{\mathbf{N}\times\mathbf{C}}$, which contains $N(= H \cdot W)$, C-dimensional vectors representing various image regions. We use learnable weight matrices $\mathbf{W_K} \in \mathbb{R}^{\mathbf{C}\times\mathbf{C}`}$, $\mathbf{W_Q} \in \mathbb{R}^{\mathbf{C}\times\mathbf{C}`}$, and $\mathbf{W_{v`}} \in \mathbb{R}^{\mathbf{C}\times\mathbf{C}`}$ to calculate key($\mathbf{K}$), query($\mathbf{Q}$) and value($\mathbf{v}`$) vectors respectively. Here $C` = C/r$, $r > 1$, r is the reduction ratio which is used to decrease the dimensions of the vectors and calculate attention weights and values in a low-dimensional subspace. Let $\mathbf{F^{N\times C}} \equiv \mathbf{F}$

$$\mathbf{K} = \mathbf{F}\cdot\mathbf{W_K}, \quad \mathbf{Q} = \mathbf{F}\cdot\mathbf{W_Q}, \quad \mathbf{v}` = \mathbf{F}\cdot\mathbf{W_{v`}} \tag{1}$$

$$\beta^{\mathbf{N}\times\mathbf{N}} = \mathbf{Q}\cdot\mathbf{K^T} \implies a_{ij} = \frac{\exp(\beta_{ij})}{\sum_{j=1}^{N}\exp(\beta_{ij})}, i,j = 1,\ldots,N \tag{2}$$

$$\mathbf{V^{N\times C`}} = \mathbf{a}\cdot\mathbf{v}` \tag{3}$$

Here, $\mathbf{a} \in \mathbb{R}^{\mathbf{N}\times\mathbf{N}}$ is the self-attention matrix, and a_{ij} is the attention weight that region i puts on region j. $\mathbf{V} \in \mathbb{R}^{\mathbf{N}\times\mathbf{C}`}$ in Eq.3 is a value matrix in the low-dimensional subspace and we use $\mathbf{W_V} \in \mathbb{R}^{\mathbf{C}`\times\mathbf{C}}$ to project it into the original subspace to get the self-attention feature maps,

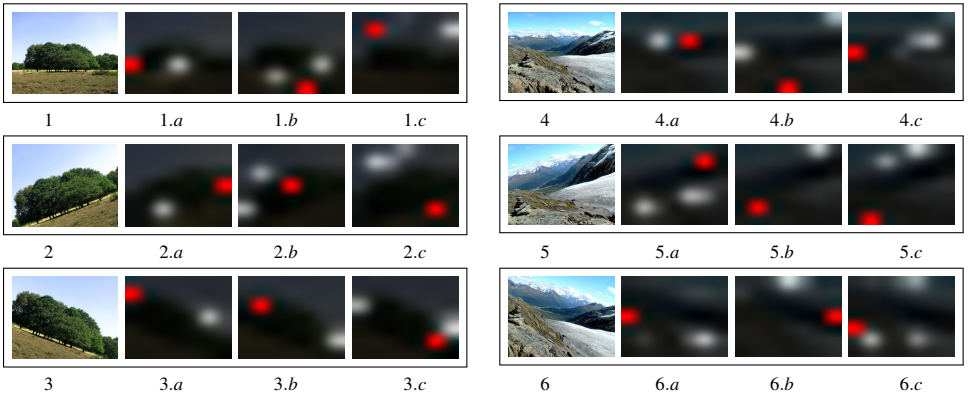


Figure 4: In the heatmaps, the red region indicates the query region and the white regions represent the points where the query region focuses its attention. Figures. 1.a, 2.a, and 3.a indicates that the query region on the horizon attends to other regions on the horizon. For the SET-B, the input images do not have straight lines but the model is able to understand the general upright orientation. In figures 4.c and 6.a, the query region on ground is attending to other ground regions. Similarly, in figures 4.a and 6.b the query region in sky is attending to another point in the sky. In figures 1.c, 2.c, and 3.c and figures 4.c, 5.c, and 6.c, the query region is attending to a far location indicating the learning of long-range dependencies.

$\mathbf{S} \in \mathbb{R}^{\mathbf{N}\times\mathbf{C}}$. Finally, we will add a residual connection to get the final output $\tilde{\mathbf{F}}^{\mathbf{N}\times\mathbf{C}}$.

$$\mathbf{S} = \mathbf{V} \cdot \mathbf{W_V} \tag{4}$$

$$\tilde{\mathbf{F}} = \mathbf{F} + \alpha\mathbf{S} \tag{5}$$

where α is a trainable scalar parameter initialized to 0. Eq.5 implies that the model first learns image features around the local neighbourhood and then gradually moves on to learn global dependencies [34]. The combination of squeeze and excite operation and spatial self-attention gives us the feature maps that are rich in content as well as contextual information.

The learning of global long-range dependencies and relative positioning of different image regions improved the model predictions on image tilt detection tasks. From the heatmaps, in Fig.4, we can see that the query (red) region is able to shift its attention with the image orientation. Specifically, in Fig.4-[1.a, 2.a, 3.a] (left column), the query (red) region on the horizon was able to focus on the other horizon points despite different orientations of the same image. In Fig.4-SET-B (right column), there is no clear horizon line but the query regions are able to attend to relevant regions for tilt detection. For example, query region on the ground in heatmaps-[4.b, 5.b] is attending to other points on the ground and query region in Fig.4-[4.a, 6.b] in sky is focusing on other points above the skyline. Also note that query regions in Fig.4-[1.c, 2.c, 3.c] (left column) are able to attend to far locations which indicates learning of long-range dependencies and **spatial information**. Therefore, for detecting image tilt angle, the neural network model needs to **learn the relative positioning of the image pixels** to differentiate between various distinct image orientations.

3.2 Image Tilt Detection

3.2.1 Motivation and Problem Modeling

Although, the intuitive approach for tackling this problem seems to be regression where we use a Deep CNN to extract image features and minimize the angular distance between the ground truth angle and the predicted angle. But training deep neural networks for regression tasks is difficult [12] and when we enter the domain of light-weight models, like MobileNetV3 or SA-MobileNet, with few parameters as compared to ResNet-50 [7] or VGG-16 [22] networks, it becomes extremely difficult to achieve good results on regression tasks. In contrast to regression, deep neural networks achieve highly accurate results on classification tasks but we cannot model the image tilt detection as a single label classification problem due to a variety of reasons. If the prediction is $1^\circ - 2^\circ$ off from the true label, the network will penalize it equally as it would if the prediction is off by $10^\circ - 20^\circ$ or more. This might not be necessary because 1° or 2° variation in the image tilt angle might not be significant. Therefore, **we train the model to predict multiple angles within a narrow interval** of the ground truth tilt angle and penalize only those values that are outside this narrow range.

3.2.2 Training Pipeline

We model the problem of Image Tilt Detection in a multi-label scenario where we train our proposed neural network model with a tilted input image and the corresponding ground truth label vector with multiple labels of tilt angles in a narrow interval of either $\pm 1^\circ$ or $\pm 2^\circ$ depending on the dataset quality. Images in the training dataset were assumed to be upright and assigned 0° ground truth label by default. Every image is rotated by each angle from $0^\circ, 1^\circ, \ldots, 359^\circ$ and center-cropped before giving to the model as input.

To enable multi-label training for predictions of image tilt angles within the narrow interval, the ground truth label vectors are constructed as 360-dimensional vectors with value of 1 at indices $G-I, \ldots, G-1, G, G+1, \ldots, G+I$, and value of 0 at rest of the indices. G and I are the ground truth image tilt angle and length of the narrow interval respectively. Note that the ground truth label vector is cyclic. If $G+x \geq 360$, for some $0 \leq x \leq I$, then we set the value 1 at index $(G+x)\%360$. Similarly, if $G-x < 0$, then we set 1 at index $G-x+360$. Last layer of the model is a 360-dimensional layer, representing all the integer angles, with sigmoid activation function that holds the prediction scores of the tilt angles for a given input image. We use Binary Cross Entropy loss function to calculate the loss between given ground truth label vector and the predicted outputs.

$$\mathcal{L}(\mathbf{y}, \mathbf{p}) = -\frac{1}{D}\left[\sum_{i=1}^{D} y_i \log p_i + (1-y_i)\log(1-p_i)\right] \tag{6}$$

Here $D = 360$ and $\mathbf{y}$ and $\mathbf{p}$ are groundtruth and the predicted vectors respectively.

4 Experiments

4.0.1 Model Prediction

For the final prediction, we take the highest scoring label, in the last layer, and if more than two labels have the highest scores, we simply take their average value. From our experiments, we saw that most of the times, when more than two labels have the highest score,

Previous Works	**Accuracy** (%) ↑	**Angle Error** (°) ↓
Ciocca *et al*. [3] (LPB-based featutes)	71.87	-
CNN + Fuzzy Edge Detection	85.21	-
Fischer *et al*. [6] (AlexNet)	-	21.23
Maji *et al*. [13] (Xception)	-	7.89
MobileNetV3 (baseline)	85.97	5.06
ResNet-50 (baseline)	**93.67**	**3.98**
SA-MobileNet (proposed)	92.39	4.27

Table 1: Accuracies and angles errors of various baseline methods on SUN397 dataset. ↑ / ↓ indicates that higher/lower is better respectively.

it was 1.0 and the corresponding labels were consecutive and belonged to set of narrow interval around the ground truth tilt angle. This implies that there is an **implicit correlation between the ground truth labeled angles** within the narrow interval that helps the model to determine the image orientation over the complete 360° range. Another advantage of using this method, over single-label classification, is that the network only penalizes those output values that are outside the narrow interval around the ground truth tilt angle. Intuition is that if the ground truth image tilt angle is 45°, then we do not want to penalize 44° or 46° prediction as much as we want to penalize 42° or 46° predictions because though they are close to the ground truth value they will not make image look upright.

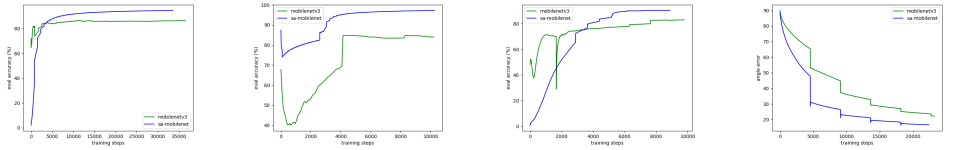
4.1 Model Architecture

For the MobileNetV3 model, the input image size is 224×224 and the subsequent layers decrease the feature map size to 7×7 in strides of 2. We integrated Spatial Self-Attention Modules within MobileNetV3 to get Self-Attention MobileNet. Spatial Self-Attention Modules were applied to the blocks of sizes 56×56, 28×28, 14×14, and 7×7. We selected these blocks because they are high-level image features which encodes meaningful image representations. Furthermore, the computational costs of calculating self-attention on these blocks was not very high because of the small feature map sizes. We also replaced the 1280-dimensional fully connected layer of MobileNetV3 with a 720-dimensional layer. This helped in reducing the MAdds that were added due to the introduction of spatial self-attention operations. As a result, the proposed SA-MobileNet model, when converted to Tflite, came out **faster by an average of 4ms** than the corresponding MobileNetV3 Tflite model when tested on Snapdragon 750 Octa core. All the convolutional kernels and fully connected layers were initialized from pretrained ImageNet [11] weights apart from the newly added self-attention blocks that were initialized randomly.

The network was trained end-to-end using RMSprop [25] optimizer with momentum 0.9. The initial learning rate was 0.001 and it was decayed using exponential learning rate schedule with $40k$ decay steps and 0.95 decay rate.

4.2 Datasets

We used publicly available SUN397 [30], ADE20K [35], and NYU-V1 [21] datasets. SUN397 is a scene understanding dataset that contains 397 well-sampled categories of diverse scenes with $108,754$ distinct images. There are 10 train-test partitions for the dataset, and for our



(a) ADE20K (b) NYU-V1 (c) SUN397 (d) Regression loss

Figure 5: Figures (a), (b), and (c) contains evaluation accuracy plots over training steps for various datasets. The accuracy curve of SA-MobileNet (blue) model is above the accuracy curve of MobileNetV3 (green) model. Fig (d) contains the plots for regression losses calculated using angle error loss function (Eq.8) and trained on ADE20K dataset.

evaluation dataset, we took the union of images from all the test partitions that resulted in 15,691 images. The remaining 92,793 images were used for training. ADE20K dataset is another scene parsing dataset with 20,210 images in the training set and 5,000 images in the evaluation set. SUN397 and ADE20K datasets contain wide variety of natural images that may or may not contain straight vertical or horizontal lines. That is why we set the interval length, $I = 2^\circ$ while training on these two datasets. We also used NYU-V1 Depth dataset that contains frames from video sequences of various indoor scenes recorded from both RGB and Depth camera of Microsoft Kinect. Before using this dataset to train our model, we had to make the images upright because all the images were skewed as seen from Fig.6. We straightened the set of 2,282 images and used 2000 images for our training and 282 images for testing. We split the data in such a way that frames from the same indoor scene does not come in both the splits. We set the interval length $I = 1^\circ$ because this dataset was manually annotated and we saw highly accurate results on the evaluation data.

4.3 Results

We train MobileNetV3 model as a baseline for mobile devices on SUN397, ADE20K and NYU-V1 dataset using the training approach, described in section 3.2. The proposed SA-MobileNet consistently performs better in terms of detection accuracies and angle errors on all the evaluation datasets as seen from Fig.5 and Table.2. We also trained MobileNetV3 and SA-MobileNet for regressing the image tilt angle, by using angle loss function, given by Eq.7 and 8, and AdaDelta optimizer on ADE20k dataset. The SA-MobileNet model gives us lower angle error when compared to the MobileNetV3 model as shown in Fig 5.d and Table 3.

$$e = \left| a_{true} - a_{pred} \right| \tag{7}$$

$$\mathcal{L}_{\textbf{angle}} = min\{e, 360 - e\} \tag{8}$$

Here a_{true} and a_{pred} are groundtruth and predicted angles respectively, with values ranging from $0^\circ - to - 359^\circ$. We also use this loss function to calculate the angle errors of the model trained with the multi-label approach. From Table.2 we see that the proposed SA-MobileNet model resulted in very low-angle errors on NYU-V1, ADE20K, and SUN397 dataset when compared with MobileNetV3 model. From the evaluation accuracy plots in Fig.5.a, Fig.5.b, Fig.5.c, we can see that the accuracy curve of our model (blue) is above the curve of MobileNetV3 model (green) during training for all the datasets. In Fig.5.d, we plot angle errors for both the models which were trained for regressing the exact orientation angle on ADE20K dataset. The SA-MobileNet model produced low angle errors when compared

Model	NYU-V1		ADE20K		SUN397	
	Acc(%) ↑	AE° ↓	Acc(%) ↑	AE° ↓	Acc(%) ↑	AE° ↓
MobileNetV3	88.02	15.79	87.68	16.84	85.97	5.06
ResNet-50	94.59	4.67	**97.84**	**3.09**	**93.67**	**3.98**
SA-MobileNet	**98.53**	**3.45**	96.77	3.45	92.39	4.27

Table 2: Evaluation accuracies and angle errors of the MobileNetV3, ResNet-50, and SA-MobileNet models on various datasets with the proposed tilt angle detection approach. Acc: Accuracy (%) and AE: Angle Errors(°). ↑ / ↓ indicates that higher/lower is better respectively.

Model	Latency(↓) (milliseconds)	Parameters(↓) (millions)
MobileNetV3	79	**4.2**
SA-MobileNet	**75**	4.5

Table 3: Tflite models were tested on Snapdragon 750, Octa core (2x 2.2 GHz, 6x 1.8 GHz) for latency measurements.

Model	Angle Error° (↓)
MobileNetV3	21.07
SA-MobileNet	**15.53**

Table 4: Regresion loss on ADE20k dataset trained with angle loss function Eq.8.

to the MobileNetV3 model. This also justifies that the **long-range dependencies learned by the self-attention modules are necessary for image tilt detection**. We also trained the ResNet-50 model as a baseline to validate the effectiveness of our training pipeline. Though, the ResNet-50 model outperforms both the MobileNetV3 and SA-MobileNet, the improvement is marginal despite the huge difference in the number of parameters between the ResNet-50 and the light-weight models.

Comparison with previous works: We also present comparison of our training approach with previous works for this problem in Table 1. Over the past few years many different methods have been proposed to tackle this problem but they have high angle errors or low accuracies. ResNet-50 baseline model gives the lowest angle error on the test dataset to give state-of-the-art results. However, the ResNet-50 network cannot be deployed on low-resource devices because it has over 25 million parameters. But the proposed Self-Attention MobileNet has around 4 million parameters with low-latency values and state-of-the-art results for mobile devices that makes it ideal to be deployed in smartphones for real-time inferences.

5 Conclusion and Future Work

We present a novel neural network model for mobile/IoT devices powered with the Self-Attention Modules. We also present highly accurate results on the task of Image Tilt Detection and are able to correct image orientations on smartphone in real-time. Our proposed training approach is also very simple but effective for this task. In the future work, we can use a dynamic value of the narrow interval, I, which can be unique for each image within limit.

We will also validate the effectiveness of Self-Attention MobileNet on other downstream tasks like image classification, object detection and image segmentation applications for

(a) 3° anticlockwise (b) 4° clockwise (c) 5° clockwise

Figure 6: Samples of images NYU-V1 dataset. Most the images(left) were skewed and had to be rotated to make them upright(right).

mobile devices.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[2] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

[3] Gianluigi Ciocca, Claudio Cusano, and Raimondo Schettini. Image orientation detection using lbp-based features and logistic regression. *Multimedia Tools and Applications*, 74(9):3013–3034, 2015.

[4] Tien Do, Khiem Vuong, Stergios I Roumeliotis, and Hyun Soo Park. Surface normal estimation of tilted images via spatial rectifier. In *European Conference on Computer Vision*, pages 265–280. Springer, 2020.

[5] Sergiy Fefilatyev, Volha Smarodzinava, Lawrence O Hall, and Dmitry B Goldgof. Horizon detection using machine learning techniques. In *2006 5th International Conference on Machine Learning and Applications (ICMLA'06)*, pages 17–21. IEEE, 2006.

[6] Philipp Fischer, Alexey Dosovitskiy, and Thomas Brox. Image orientation estimation with convolutional networks. In *German Conference on Pattern Recognition*, pages 368–378. Springer, 2015.

[7] 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.

[8] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Jonathan Eisenmann, Matthew Fisher, Emiliano Gambaretto, Sunil Hadap, and Jean-François Lalonde. A perceptual measure for deep single image camera calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2354–2363, 2018.

[9] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019.

[10] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[11] 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.

[12] Stéphane Lathuilière, Pablo Mesejo, Xavier Alameda-Pineda, and Radu Horaud. A comprehensive analysis of deep regression. *IEEE transactions on pattern analysis and machine intelligence*, 42(9):2065–2081, 2019.

[13] Subhadip Maji and Smarajit Bose. Deep image orientation angle detection. *arXiv preprint arXiv:2007.06709*, 2020.

[14] Greg Olmschenk, Hao Tang, and Zhigang Zhu. Pitch and roll camera orientation from a single 2d image using convolutional neural networks. In *2017 14th Conference on Computer and Robot Vision (CRV)*, pages 261–268. IEEE, 2017.

[15] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*, 2016.

[16] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Bam: Bottleneck attention module. *arXiv preprint arXiv:1807.06514*, 2018.

[17] Master Prince, Suliman A Alsuhibany, and Nahid A Siddiqi. A step towards the optimal estimation of image orientation. *IEEE Access*, 7:185750–185759, 2019.

[18] C Reshmalakshmi and M Sasikumar. Image edge orientation estimation via fuzzy logic. *Materials Today: Proceedings*, 4(2):4274–4282, 2017.

[19] Rajat Saini, Nandan Kumar Jha, Bedanta Das, Sparsh Mittal, and C Krishna Mohan. Ulsam: Ultra-lightweight subspace attention module for compact convolutional neural networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1627–1636, 2020.

[20] 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.

[21] Nathan Silberman and Rob Fergus. Indoor scene segmentation using a structured light sensor. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 601–608. IEEE, 2011.

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[23] Hungsun Son and Kok-Meng Lee. Two-dof magnetic orientation sensor using distributed multipole models for spherical wheel motor. *Mechatronics*, 21(1):156–165, 2011.

[24] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[25] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[26] A Waswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, AN Gomez, L Kaiser, and I Polosukhin. Attention is all you need. In *NIPS*, 2017.

[27] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[28] Scott Workman, Menghua Zhai, and Nathan Jacobs. Horizon lines in the wild. *arXiv preprint arXiv:1604.02129*, 2016.

[29] Wenqi Xian, Zhengqi Li, Matthew Fisher, Jonathan Eisenmann, Eli Shechtman, and Noah Snavely. Uprightnet: Geometry-aware camera orientation estimation from single images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9974–9983, 2019.

[30] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[31] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[32] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.

[33] Menghua Zhai, Scott Workman, and Nathan Jacobs. Detecting vanishing points using global image context in a non-manhattan world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5657–5665, 2016.

[34] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019.

[35] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.