

A light-weight Deep Learning Model for Hand Gesture Recognition in space applications

Sandeep Nithyanandan¹, Chithra A V², Somoshree Datta³, and Manusubramanian S⁴

¹ sandeepnithyanandan@lpssc.gov.in

² av_chithra@lpssc.gov.in

³ somoshreedatta@lpssc.gov.in

⁴ manuss@lpssc.gov.in

Indian Space Research Organisation, Thiruvananthapuram, Kerala

Abstract. Hand gesture recognition is a field of research involving computer vision and artificial intelligence to recognize and interpret hand gestures. Based on this interpretation, systems can be built and configured to perform various actions as needed. The research that has been carried out till date regarding the recognition of hand gestures involving space applications is very limited. Hence, we try to address this field of research whereby it is imperative to have accurate recognition of gestures for controlling space systems. The astronauts on board operate in very constrained environments. Both RGB and thermal data play vital roles in space exploration and enable astronauts to make informed decisions, troubleshoot problems, and enhance their understanding of the space environment. Hence, it is necessary to innovate and design precise AI models those will help in the smooth functioning of space systems. Moreover, in space missions, where resources like power, memory, and computational capacity are limited, it is crucial to have lightweight machine learning or deep learning models onboard. The architectural model that we propose in this paper is a light weight deep learning model which works on both RGB and Thermal modalities with better accuracy as compared to the state-of-the-art research. Our dual-mode model has achieved an accuracy of 93.27% on RGB images and 82.48% on thermal images respectively.

Keywords: Hand gesture recognition · lightweight · deep learning · deep CNN

1 Introduction

Hand gesture recognition is a flourishing field of research in Computer Vision and Artificial Intelligence which aims to develop AI models and algorithms that can interpret hand gestures (including both human-made gestures as well as robotic gestures) and perform essential actions based on this interpretation. The goal of this field of research is to enable machines to recognize and respond to gestures in a natural and intuitive way, which can have significant applications in areas

such as Human-Computer Interaction (HCI), robotics, space applications, etc. amongst many more.

The major focus in this area of research is to develop *real-time* algorithms and models that can detect and recognize hand gestures accurately. In this paper, we have mainly focused on the use-case of the application of hand gesture recognition in space exploration and operations. In space, astronauts operate in a confined and challenging environment where using keyboards or touchscreens may not be practically reliable and feasible. There is a need to collaborate with robots, certain agents as well as systems to accomplish a mission smoothly. Hand gesture recognition can provide a much more intuitive and natural way for astronauts to interact with operational systems, which can lead to improved efficiency. The model that we propose in this paper is not only applicable to humans but also to AI assistants. Moreover, hand gesture recognition will also enable astronauts to communicate in a clear and unambiguous way in situations where spoken communication might be difficult or impossible, such as noisy or high-risk environments. Overall, the potential that one can harness out of hand gesture recognition in space applications is immense. It can greatly enhance human-machine interaction in space and act as a valuable tool for astronauts operating in this challenging environment. Since the environment of operation of this tool is pretty restricted, lightweight hand gesture models are preferred for ease of implementation and use.

Space missions often use both RGB and thermal cameras for various purposes. These cameras are typically mounted on the spacecraft, space station, or on specialized equipment such as robotic arms or scientific payloads. RGB cameras are used to capture color images of objects or scenes in space. Thermal cameras, on the other hand, are used to capture images that depict the distribution of temperature variations in a scene. In some cases, space missions may use cameras that combine both RGB and thermal capabilities, allowing for the simultaneous capture of color and thermal images.

2 Key Contributions

The major contributions of this paper are given below:

- **A Dual-mode Hand Gesture Recognition Algorithm:** As specified in the section above, our proposed algorithm works well on both RGB as well as thermal images.
- **A light-weight model for hand gesture recognition in space applications:** As our concern is on building a sustainable, light-weight AI model for space activities, we have tried to strike a good balance between the number of parameters involved and the accuracy of our model.

3 Related Work

This section briefly reviews the relevant literature of machine learning and deep learning models used for recognizing hand gestures. With each passing day, the

demand with respect to the level of services required by humans is increasing. When communicating face-to-face, hand gestures are a key element. Hence, for accurate communication, it is absolutely necessary to interpret and find out the real meaning behind any hand gesture in order to respond back. Till date, various machine learning models and classifiers have been used for hand gesture recognition, depending on the specific application and requirements that the task demands. In order to perform a prediction, hand gesture images are first fed into the system, followed by further analysis which is usually carried out by applying machine learning algorithms as stated in the paper [1]. State-of-the-art algorithms such as CNN, Bagging, Support Vector Machines (SVMs) and Hidden Markov Models (HMM) have been used in the past and have provided satisfactory results. Also, a lot of work has been done on using depth for hand tracking and gesture recognition. Depth sensors such as Kinect from Microsoft are widely used for hand tracking applications, though the primary focus of Kinect systems is more on the applications rather than on localization and classification techniques as shown in paper [2].

A real-time hand gesture recognition system has also been explored and developed by the Tam et. al [3] where they have used an embedded convolutional neural network for classification of hand-muscle contractions that are felt in the forearm. The authors were able to greatly enhance the reliability, security and response times of their AI system by using hand prosthesis that leveraged HD-EMG and deep learning. Allard et. al [4] has leveraged the potential of deep learning algorithms by applying transfer learning on an aggregated dataset collected from multiple users. Using deep learning calculations in complex tasks has become progressively much more popular due to their unmatched potential of being able to simultaneously process multiple features from huge amounts of information. Moreover, Convolutional Neural Networks (CNNs) are becoming particularly popular for recognition problems due to their inherent characteristics of identifying patterns in visual data. Jayadeep et. al [5] have proposed the design of an Indian Sign Language (ISL) hand gesture motion translation tool for banks in order to help the deaf and mute community to convey their thoughts and needs by translating them into text. Their proposed methodology recognizes the various hand gestures by considering the various secluded dynamic Indian signs that are associated with banks. The hand gesture movements were identified from a series of video frames and a Convolutional Neural Network named Inception v3 was used for feature extraction from these images. Finally, the gestures were categorized and translated into text using Long Short Term Memory (LSTM), thereby providing an accurate and promising technique for the mute and deaf community to communicate.

The architecture that we propose in this paper has leveraged the inherent characteristics of a CNN for feature extraction from traditional RGB images as well as from thermal images. From the point of view of space applications, it is extremely necessary for our proposed model to perform well and give accurate predictions for hand gesture movements in a well-lit background as well as gesture movements made in dark or not so well-lit environment. Hence, in

this paper, we propose a hand-gesture recognition model, which is a type of deep Convolutional Neural Network (addressed in this paper as *deep CNN*), to achieve state-of-the-art results which not only gives accurate predictions for gestures made in illuminated environment, but also for thermal images captured in low-light or completely dark environments.

4 Methodology

In this section, we discuss our proposed architecture in a detailed manner. Since our proposed architecture is also a deep Convolutional Neural Network (CNN), so we give a brief idea about the same before moving forward to discussing our proposed work.

4.1 Deep CNN

Deep CNNs (Convolutional Neural Networks) are a type of neural network that are specifically designed for image and video recognition tasks. They are composed of multiple layers of convolutional and pooling operations, which are used for feature extraction from the input image and finally identifying the patterns that are relevant to the task at hand.

The term "*deep*" in deep CNNs refers to the fact that they have many layers, typically ranging from tens to hundreds of layers which enables these architectures to learn more complex features and thereby achieve higher accuracy as compared to shallow CNNs. It has been proven in the past that deep CNNs are highly effective for a wide range of computer vision tasks such as image classification, object detection, semantic segmentation and many more. Deep CNNs work by passing the input image through a series of convolutional layers, which convolve the image with a set of learned filters to extract features such as edges, corners and textures. The output obtained from each convolutional layer is then passed through a non-linear activation function, such as ReLU (Rectified Linear Unit) for example, which introduces non-linearity into the model and enables it to recognize more complex features from the input images. After passing the images through multiple convolutional and activation operations, under-sampling of the output is done through a pooling layer which reduces the spatial dimensions of the feature map but preserves the most important features identified. Finally, the output of the last convolutional layer is flattened and passed through one or more connected layers, which combines the extracted features into a high-level representation that can be used for classification, object detection or other tasks.

4.2 Proposed work

In this paper we propose a light weight dual-mode Hand Gesture Recognition Model which works on both RGB and thermal data based on the input domain

(See Figure 1). The input images undergo initial processing through a *Mode Selector* module. The *Mode Selector* module activates either the RGB or Thermal mode based on the nature of the input data. It differentiates between the RGB and thermal images by analyzing their color channels. Standard RGB images comprise of three colour channels: red, green and blue, while standard thermal images are typically composed of a single grayscale channel representing temperature. The images are further pre-processed and their dimensions are changed to align with the dimensions expected by our model. The pre-processed data is then fed into a deep Convolutional Neural Network (Deep CNN) architecture. The layers of the deep CNN architecture extract relevant features from the data, and make predictions for the gestures based on those extracted features.

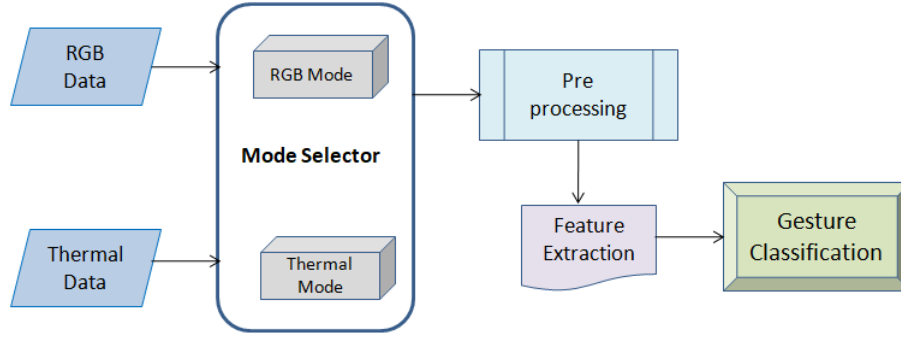


Fig. 1. Proposed Architecture: A Dual Mode Hand Gesture Recognition Algorithm which detect gestures of hands in both RGB and Thermal data

Model Architecture: A deep convolutional neural network is used for extracting features and recognizing the hand gestures. It consists of a series of building blocks, each of which contains three layers.

The first layer is a **1x1 convolutional layer with ReLU activation function**. This layer is responsible for reducing the number of input channels to the subsequent layers while also introducing non-linearity. The ReLU activation function is used to ensure that the output of the layer remains within a range of 0 to 6, which helps prevent the gradient from vanishing or exploding during training.

$$y = \text{ReLU}(Wx + b) \quad (1)$$

where x is the input tensor with dimensions [batch_size, height, width, input_channels], W is the weight tensor with dimensions [1, 1, input_channels, output_channels], b is the bias tensor with dimensions [output_channels], and $\text{ReLU}()$ is the ReLU activation function defined as:

$$\text{ReLU}(x) = \min(\max(x, 0), 6) \quad (2)$$

The output y has the same spatial dimensions as the input but with `output_channels` as number of channels.

The second layer is a **depthwise convolutional layer**, which applies a separate convolutional filter to each input channel. This allows the network to learn more efficient representations by reducing the number of computations required. Depthwise convolutional layers have fewer parameters than traditional convolutional layers and can be more computationally efficient. The output of this layer has the same spatial dimensions as the input but with a reduced number of channels.

$$y = \text{ReLU}(\text{depthwise}(Wx) + b) \quad (3)$$

where $\text{depthwise}()$ is the depthwise convolution operation that applies a separate 3×3 convolutional filter to each input channel, W is the weight tensor with dimensions $[3, 3, \text{input_channels}, \text{depth_multiplier}]$, depth_multiplier is a hyper-parameter that controls the number of output channels per input channel, b is the bias tensor with dimensions $[\text{depth_multiplier} * \text{input_channels}]$, and $\text{ReLU}()$ is the ReLU activation function. The output y has the same spatial dimensions as the input but with $\text{depth_multiplier} * \text{input_channels}$ number of channels.

The third layer is a **1×1 convolutional layer without any non-linearity**. This layer is responsible for increasing the number of channels back to the original number. By performing a 1×1 convolution, the layer is able to mix the features from the different channels in a way that is learned during training. The lack of non-linearity allows the layer to preserve more information about the input, which can be important for tasks that require fine-grained details.

$$y = Wx + b \quad (4)$$

where W is the weight tensor with dimensions $[1, 1, \text{input_channels} * \text{depth_multiplier}, \text{output_channels}]$, b is the bias tensor with dimensions $[\text{output_channels}]$, and x is the input tensor with dimensions $[\text{batch_size}, \text{height}, \text{width}, \text{input_channels} * \text{depth_multiplier}]$. The output y has the same spatial dimensions as the input but with `output_channels` number of channels.

In summary, the first three layers of the CNN work together to reduce the number of input channels, apply depthwise convolution, and increase the number of output channels. This approach helps to reduce computational complexity while also preserving important information about the input.

The penultimate layer is a **Global max pooling 2D** layer which is a type of pooling layer commonly used in convolutional neural networks (CNNs). It operates on the output of a convolutional layer or a set of convolutional layers and is typically used as a way to reduce the dimensionality of the feature maps before feeding them to the fully connected layers.

Given a feature map or activation tensor with dimensions (C, H, W) , where C is the number of channels, and H and W are the height and width of the feature map, the global max pooling operation takes the maximum value of each channel across the entire spatial dimension, resulting in a 1D output.

For each channel c , the global max pooling operation is defined as:

$$max_pooling_c = max_{i,j}(featureMap_c_{ij}) \quad (5)$$

where $max_pooling_c$ denotes the output value of the global max pooling 2D operation for a channel c , $max_{i,j}$ operation computes the maximum value within the spatial slice of $[i,j]$ and $featureMap_c_{ij}$ denotes the activation value at spatial location (i, j) in channel c .

The global max pooling 2D layer has several advantages over other pooling layers such as average pooling. First, it preserves the most salient features of the input and discards the irrelevant ones. Second, it is less affected by translation invariance, which is an important property of CNNs. Finally, it reduces the number of parameters and computations in the network, which can help prevent overfitting and reduce training time.

The final layer is **fully connected dense layer with softmax activation**, also known as a fully connected layer or a dense layer. Mathematically, the output of the dense layer with softmax activation can be represented as follows:

$$output = softmax(W * input + b) \quad (6)$$

where $input$ represents the input to the layer, which is typically the output of the previous layer, W is the weight matrix connecting each neuron in the current layer to each neuron in the subsequent layer (output layer), b is the bias vector added to each neuron in the output layer and $softmax()$ is the softmax activation function, which transforms the raw output values into a probability distribution over the gestures.

5 Experiments and Results

In this section, we discuss about the datasets used to conduct the experiments and the final results that are obtained.

5.1 Dataset

The proposed architectural model has been trained and tested on Arabic Sign Language (ArSL) dataset [6] for RGB images and 'Plain Background Thermal Imaging Dataset' for Thermal images [7]. A sample of the ArSL dataset is shown in Figure 2.

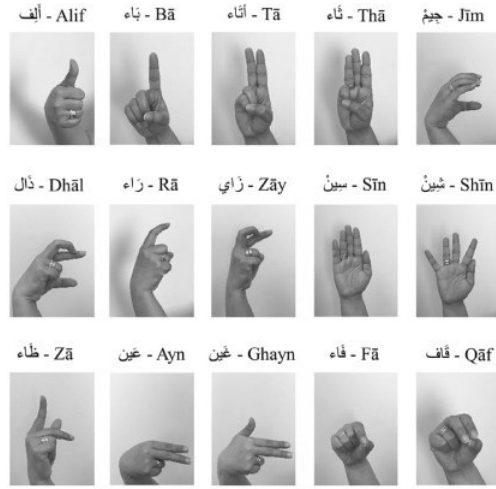


Fig. 2. Sample data from ArSL Dataset

The ArSL dataset contains 34,455 images of Arabic sign language hand gesture representations that has been performed by more than 40 people for 32 standard Arabic signs and alphabets. But, for our purpose, we have taken only 20 out of the 32 classes. The number of images per class differs from one class to another. 75% of the images have been used for training, whereas the remaining 25% has been used for testing the model. The initial dimensions of the image were 64x64. Similarly, the Plain background thermal imaging dataset of hand gestures comprises of 3600 images in total. There are two kinds of images in this dataset - grayscale and color scale images. There are a total of 10 classes of hand gestures included in this dataset. A sample of the thermal image dataset is shown in Figure 3.

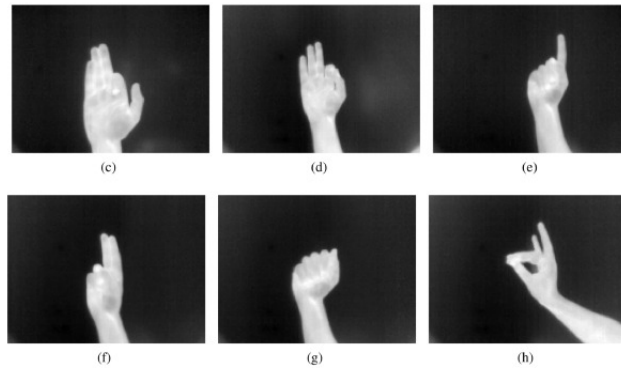


Fig. 3. Sample data from Thermal Dataset of Hand Gestures

The model has been trained separately on the thermal dataset by taking 2700 images and keeping aside the remaining 900 images for testing. The images used are of 160x120 dimensions.

Experimental setup: The network was trained and tested with Arabic Sign Language (ArSL) dataset and Plain background thermal imaging dataset of hand gestures. Our proposed model has been trained for 25 epochs. The batch size was 16 and the optimizer was Adam. The input image dimensions were set to 224x224x3.

5.2 Performance Evaluation:

Performance evaluation of our proposed model has been carried out with the measures of accuracy, precision, recall, f1-score and support. Confusion matrices were also drawn to analyze the class-wise predictions. We compare our proposed model with MobileNetV2 [8] model as benchmark. MobileNetV2 is a CNN architecture designed by Google to provide efficient and accurate image classification on mobile and embedded devices. MobileNetV2 is an improvement over the original MobileNet model, incorporating depthwise separable convolutions, linear bottlenecks, and other techniques to achieve higher efficiency and better performance. The benchmark model gave an accuracy of 90.01% on ArSL dataset and 80.22% on Plain background thermal imaging dataset respectively. Experiments were conducted on our proposed model by using both RGB and thermal image datasets. Our proposed light-weight model gave 93.27% accuracy on ArSL dataset and 82.48% accuracy on Plain background thermal imaging dataset. Also, our proposed model has 3.3 million trainable parameters, as compared to 3.4 million trainable parameters of MobileNetV2, thus making our model light-weighted as compared to our benchmark model. The Confusion Matrix pertaining to thermal dataset is shown in Fig: 4 and the other performance measures related to thermal dataset are shown in Table 1.

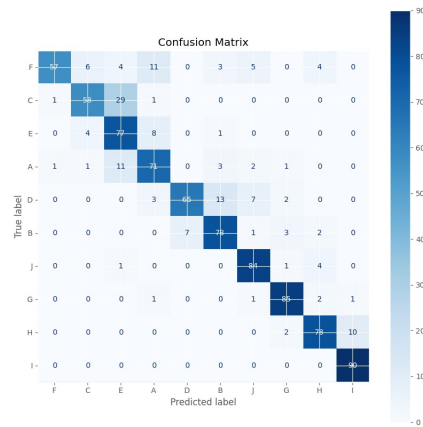


Fig. 4. Confusion Matrix (Thermal Mode): 82.48% accuracy on Thermal Dataset

Table 1. Performance Evaluation on Thermal Dataset

Class	Precision	Recall	F1 Score	Support
A	0.97	0.63	0.77	90
B	0.84	0.65	0.73	89
C	0.63	0.86	0.73	90
D	0.75	0.79	0.77	90
E	0.90	0.72	0.80	90
F	0.80	0.86	0.83	91
G	0.84	0.93	0.88	90
H	0.90	0.94	0.92	90
I	0.87	0.87	0.87	90
J	0.89	1.00	0.94	90

Similarly, the Confusion matrix pertaining to RGB dataset is shown in Fig: 5 and the other performance measures related to RGB dataset are shown in Table 2.

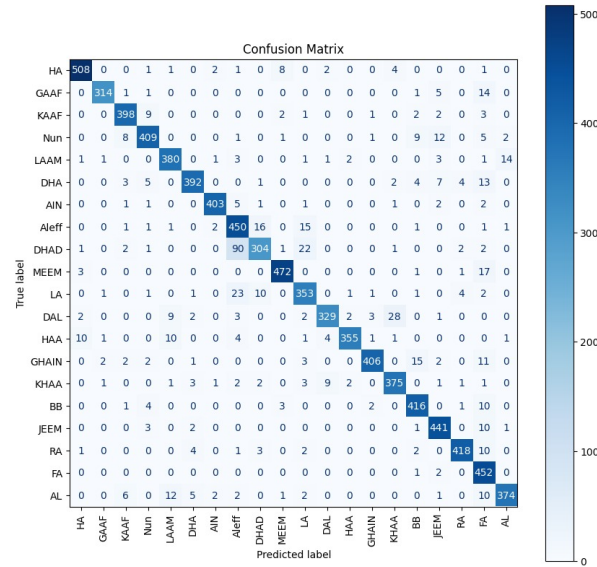
**Fig. 5. Confusion Matrix (RGB Mode):** 93.27 % accuracy on ArSL Dataset

Table 2. Performance Evaluation on RGB Dataset

Class	Precision	Recall	f1 score	Support
BB	0.96	0.96	0.96	528
KHAA	0.96	0.94	0.95	336
Nun	0.92	0.96	0.94	418
DAL	0.94	0.92	0.93	448
KAAF	0.91	0.93	0.92	408
HAA	0.97	0.90	0.94	431
RA	0.99	0.95	0.97	417
AL	0.76	0.93	0.84	489
GHAIN	0.90	0.70	0.79	426
JEEM	0.97	0.95	0.96	494
DHAD	0.86	0.88	0.87	398
LA	0.96	0.86	0.91	381
FA	0.98	0.92	0.95	388
DHA	0.98	0.90	0.94	444
LAAM	0.91	0.93	0.92	402
GAAF	0.93	0.94	0.94	437
Aleff	0.91	0.95	0.93	458
HA	0.96	0.94	0.95	441
AIN	0.77	0.99	0.87	455
MEEM	0.96	0.88	0.92	415

6 Conclusion

In this paper, we propose a dual mode, light-weight hand gesture recognition system for space applications. Our proposed work shows promising results in hand gesture recognition due to its ability to extract complex features from input data and high accuracy. Thermal images are also becoming popular due to their robustness and ability to work in low light conditions. However, there is still much research that is to be done in this field, particularly in increasing the accuracy of this algorithm as well as making it more robust to variations in lighting conditions, hand poses and other factors. Overall, it is visible that hand gesture recognition has the potential to greatly enhance human-machine interaction and improve accessibility for people working in constrained environments.

References

1. Bhushan, S., Alshehri, M., Keshta, I., Chakraverti, A. K., Rajpurohit, J., and Abugabah, A., “An experimental analysis of various machine learning algorithms for hand gesture recognition,” *Electronics* **11**(6) (2022).
2. Suarez, J. and Murphy, R. R., “Hand gesture recognition with depth images: A review,” in [2012 *IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*], 411–417 (2012).
3. Tam, S., Boukadoum, M., Campeau-Lecours, A., and Gosselin, B., “A fully embedded adaptive real-time hand gesture classifier leveraging hd-semg and deep learning,” *IEEE Transactions on Biomedical Circuits and Systems* **14**, 232–243 (2019).
4. Côté-Allard, U., Fall, C. L., Drouin, A., Campeau-Lecours, A., Gosselin, C., Glette, K., Laviolette, F., and Gosselin, B., “Deep learning for electromyographic hand gesture signal classification using transfer learning,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **27**(4), 760–771 (2019).
5. Jayadeep, G., Vishnupriya, N., Venugopal, V., Vishnu, S., and Geetha, M., “Mudra: Convolutional neural network based indian sign language translator for banks,” in [2020 *4th International Conference on Intelligent Computing and Control Systems (ICICCS)*], 1228–1232 (2020).
6. Latif, G., Mohammad, N., Alghazo, J., AlKhalaf, R., and AlKhalaf, R., “Arasl: Arabic alphabets sign language dataset,” *Data in brief* **23**, 103777 (2019).
7. Yeduri, S. R., Breland, D. S., Pandey, O. J., and Cenkeramaddi, L. R., “Updating thermal imaging dataset of hand gestures with unique labels,” *Data in Brief* **42**, 108037 (2022).
8. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C., “Mobilenetv2: Inverted residuals and linear bottlenecks,” in [Proceedings of the *IEEE conference on computer vision and pattern recognition*], 4510–4520 (2018).