# Multistage Inferencing Approach on Large Datasets in Enhancing STEM Education

Ivan Jie Xiong Ang* and King Hann Lim

*Department of Electrical and Computer Engineering,*
*Curtin University Malaysia,*
*CDT 250, 98009, Miri, Sarawak, Malaysia*

Large datasets training with deep learning neural network are often tedious and require significant amount of computational power, memory space and time. This paper presents multistage inferencing approach in deep learning neural network models when training large datasets. Datasets are arranged in heirarchical order. Subsequently, saperate models are trained for each class and subclass. Inferencing is then done in multiple stages whereby the general object class is first determined before moving forward to identify its specific subclass. A recognition rate of 90.68% is obtained after the multistage inferencing approach is applied on large STEM datasets. It was a 3.68% increment as compared to the traditional all-in-one network, which had a recognition rate of 87%. This approach is implemented onto a mobile application named as AUREL (Augmented Reality Learning). AUREL uses image recognition to detect an object and then displays the object in Augmented Reality (AR). This AR visualization is used to improve the understanding of STEM subjects and increases the enthusiasm of students towards STEM subjects.

**Keywords:** multistage inferencing approach; large dataset; augmented reality; STEM education

## I. INTRODUCTION

Enhancing STEM education is important to inspire the young generation to always stay curious about the universe. However, the current education system is highly dependent on textbooks, which is ineffective in conveying information (Xie *et al.*, 2015). As a consequence, students easily lose interest in STEM subjects due to the lack of interactive and intriguing activities. The Flipped Classroom Model (Tucker, 2012) is one of the modern approaches in teaching and learning sector. It engages learners to study effectively through the use of technology in classroom activities. To learn STEM using mobile application, an effective computer vision algorithm for object classification is highly required to detect images on textbooks automatically. Subsequently, it retrieves the corresponding information from cloud database to arose students' curiosity.

Convolutional Neural Networks (CNN) (LeCun *et al.*, 1998) is one of the recent object classification approaches due to its high accuracy in detecting and classifying objects. CNN contains multiple convolutional layers to extract features. AlexNet (Krizhevsky *et al.*, 2012) is an improved version of CNN architecture which has won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky *et al.*, 2015) by reducing the top-error to only 15.3%. GoogLeNet (Szegedy *et al.*, 2015), also known as Inception, is a well-known deep CNN architecture. It won ILSVRC 2014 by achieving a top-5 error rate of 6.67% which is very near to human level performance.

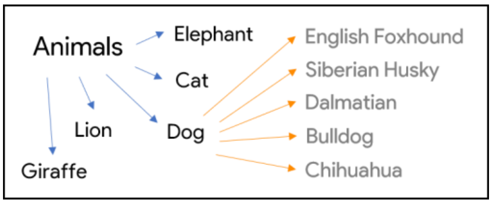


Figure 1. Hierarchical order of dataset

*Corresponding author's e-mail: ivanangjx@gmail.com

Table 1. MobileNets Body Architecture

| Type / Stride | Filter Shape | Input Size |
|---|---|---|
| Conv / s2 | 3 × 3 × 3 × 32 | 224 × 224 × 3 |
| Conv dw / s1 | 3 × 3 × 32 *dw* | 112 × 112 × 32 |
| Conv / s1 | 1 × 1 × 32 × 64 | 112 × 112 × 32 |
| Conv dw / s2 | 3 × 3 × 64 *dw* | 112 × 112 × 64 |
| Conv s1 | 1 × 1 × 64 × 128 | 56 × 56 × 64 |
| Conv dw / s1 | 3 × 3 × 128 *dw* | 56 × 56 × 128 |
| Conv / s1 | 1 × 1 × 128 × 128 | 56 × 56 × 128 |
| Conv dw / s2 | 3 × 3 × 128 *dw* | 56 × 56 × 128 |
| Conv / s1 | 1 × 1 × 128 × 256 | 28 × 28 × 128 |
| Conv dw / s1 | 3 × 3 × 256 *dw* | 28 × 28 × 256 |
| Conv / s1 | 1 × 1 × 256 × 256 | 28 × 28 × 256 |
| Conv dw / s2 | 3 × 3 × 256 *dw* | 28 × 28 × 256 |
| Conv / s1 | 1 × 1 × 256 × 512 | 14 × 14 × 256 |
| 5 × Conv dw / s1<br>Conv / s1 | 3 × 3 × 512 *dw*<br>1 × 1 × 512 × 512 | 14 × 14 × 512<br>14 × 14 × 512 |
| Conv dw / s2 | 3 × 3 × 512 *dw* | 14 × 14 × 512 |
| Conv / s1 | 1 × 1 × 512 × 1024 | 7 × 7 × 512 |
| Conv dw / s2 | 3 × 3 × 1024 *dw* | 7 × 7 × 1024 |
| Conv / s1 | 1 × 1 × 1024 × 1024 | 7 × 7 × 1024 |
| Avg Pool / s1 | Pool 7 × 7 | 7 × 7 × 1024 |
| FC / s1 | 1024 × $n$ | 1 × 1 × 1024 |
| Softmax / s1 | Classifier | 1 × 1 × $n$ |

Recent breakthroughs in deep learning neural networks has been attributed to the growing size of machine learning datasets. High quality large datasets such as ImageNet (Deng *et al*., 2009), Common Objects in Context (COCO) (Lin *et al*., 2014) and Open Images (Kuznetsova *et al*., 2018) are publicly available to advance research in the field of computer vision. Large dataset is mainly required for object classification in STEM learning to detect a wide array of objects. Often, datasets are too large to be able to fit in memory while training deep learning approaches.

In this paper, multistage inferencing approach is proposed to solve the issues encountered while training on large datasets. As shown in Figure 1, by splitting up large datasets into multiple smaller ones and arranging them in hierarchical order, more accurate results can be achieved in every subclass. Additional classes can also be easily added sequentially by performing transfer learning onto the sub-models. The proposed multistage inferencing approach is implemented onto a mobile application called AUREL (Augmented Reality Learning) (Ang & Lim, 2019) which uses object classification to detect objects and subsequently displays the object in Augmented Reality (AR). The outline of the paper is: Section II introduces the basic architecture of MobileNets followed by the proposed multistage inferencing approach. Section IV describes the implementation of mobile application on Google Cloud Platform followed by the results and discussion.

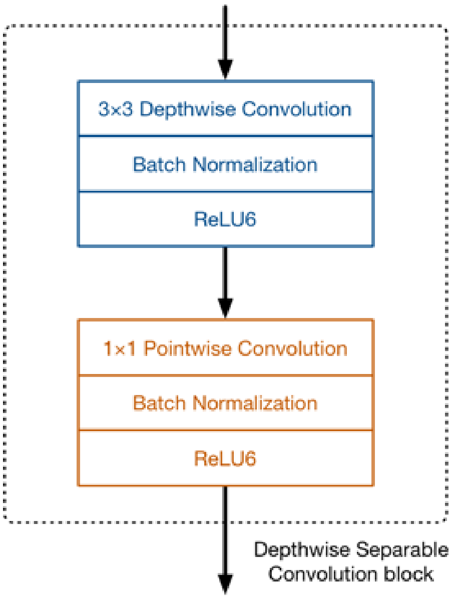
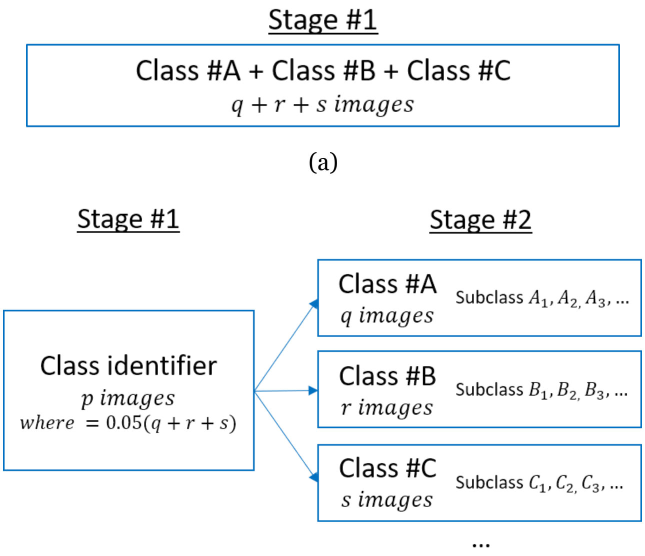

3×3 Depthwise Convolution
Batch Normalization
ReLU6
1×1 Pointwise Convolution
Batch Normalization
ReLU6
Depthwise Separable Convolution block

Figure 2. Depthwise Separable Convolution Block (Howard *et al*., 2017)

## II. MOBILENET ARCHITECTURE

Deep learning neural network architectures such as AlexNet (Krizhevsky *et al*., 2012), Inception-V3 (Szegedy *et al*., 2017) and MobileNets (Howard *et al*., 2017) are popular for mobile applications. A balance between resource constraints, speed and time is important in the miniature nature of mobile devices (Wang *et al*., 2012). Therefore, MobileNets for image classification is highly recommended in mobile applications due to its strength in minimizing time and space for image classification while only compromising the accuracy slightly. The architecture of MobileNets uses depthwise separable convolutions (Conv dw) as shown in Figure 2 instead of commonly used convolutional layers. The depthwise separable convolution block is split into two parts: first the depthwise convolution layer filters the input, then the pointwise convolution layer combines these filtered values to create new features. The implementation of depthwise separable convolution significantly reduces the number of parameters compared to the same depth networks with normal convolution operation. This causes the reduction in total number of floating point multiplication operations which is favorable in mobile and embedded applications with less computation power.


Stage #1
Class #A + Class #B + Class #C
$q + r + s$ images
Stage #1
Stage #2
Class identifier
$p$ images
where $= 0.05(q + r + s)$
Class #A
$q$ images
Subclass $A_1, A_2, A_3, \ldots$
Class #B
$r$ images
Subclass $B_1, B_2, B_3, \ldots$
Class #C
$s$ images
Subclass $C_1, C_2, C_3, \ldots$
...

(a)

(b)

Figure 3. (a) All-in-one Network (b) Multistage Inferencing Approach

A depthwise convolution with 1 filter per input channel (input depth) can be written as:

$$\hat{G}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} \times F_{k+i-1,l+j-1,m} \qquad (1)$$

where $\hat{K}$ is the depthwise convolutional kernal size of i × j × $m$ with $m_{th}$ filtering mask. The $\hat{K}$ kernel is applied to the $m_{th}$ channel in F to produce the $m_{th}$ channel of the filtered output feature map $\hat{G}$.

The full architecture of MobileNets consists of a 3×3 convolution as the first layer, followed by 13 times the depthwise separable convolution block as listed in Table 1. With an input image of 224×224×3, the output of the network will produce a 7×7×1024 feature map. The average pooling layer then reduces the spatial resolution to one before moving forward to the fully connected layer which contains all the additional parameters. The total number of classes in the model is given by $n$, and will be used in both the fully connected layer and softmax layer. The 1-dementional feature vector is then fed into the softmax layer for classification. Softmax layer maps the non-normalized output and produces a vector that represents the probability distribution a list of potential detected objects. The class with the highest probability score will be chosen and proceed to the next stage. The softmax function is defined as:

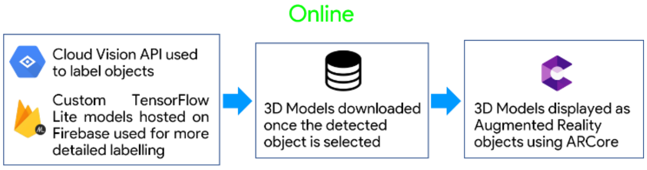
$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} for\ i = 1, \ldots, K. \qquad (2)$$

where K is the real numbers in the input vector (z) and $z_i$ is the input element. After applying the standard exponential function to each element $z_i$ of the input vector z, these values are then normalized by dividing by the sum of all these exponentials. The sum of all components of the output vector will add up to 1 after softmax function is applied.

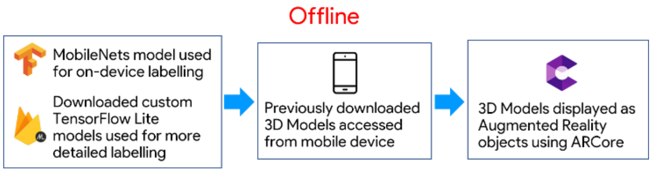
## III. MULTISTAGE INFERENCING APPROACH

In a typical training procedure, all object images will be used to train a deep learning neural network model. To detect a wide spectrum of objects, a large dataset is highly required to train a model. Training deep learning algorithms on large datasets often requires large volumes of computer memory to hold these large amount of data files. Reducing the number of training images per class may result in the loss of accuracy. Inspired by the hierarchical structure of ImageNet (Deng *et al.*, 2009), multistage inferencing approach is introduced to reduce the memory requirements while maintaining the desired accuracy.

Normally, when training an All-in-one Network, all subclasses are trained together to form a single model as seen in Figure 3(a). Instead of training one huge model which spans all classes and subclasses, a model for each of these classes and subclasses is trained by using the multistage inferencing approach. The datasets can be split into hierarchical order and each subset can be trained independently as depicted in Figure 3(b). First, a class identifier is trained using p images. This dataset is made up of 5% of the images of each subclass within the classes to represent each class. The trained model will be used in Stage #1 to identify between the subclasses. Next, separate models are trained for each of the classes using the 95% remaining images in each class to identify the subclasses. One of these models will be selected and used in the second


Online
Cloud Vision API used to label objects
Custom TensorFlow Lite models hosted on Firebase used for more detailed labelling
3D Models downloaded once the detected object is selected
3D Models displayed as Augmented Reality objects using ARCore

(a)


Offline
MobileNets model used for on-device labelling
Downloaded custom TensorFlow Lite models used for more detailed labelling
Previously downloaded 3D Models accessed from mobile device
3D Models displayed as Augmented Reality objects using ARCore

(b)

Figure 4. Overview of AUREL system during (a) online and (b) offline situations

stage of the multistage inferencing approach depending on the results of the first stage.

Since the total number of images used in both the all-in-one network and multistage inferencing approach are the same, the time taken to complete an epoch during training on both methods are similar. However, multistage inferencing approach can ease and benefit the process of adding subclasses in the future via Transfer Learning. When additional classes are inserted to an all-in-one network, the entire large dataset has to be retrained which takes a considerable amount of memory space and time. Whereas in the multistage inferencing approach, for example, if a subclass #B4 is to be inserted into Class #B as seen in Figure 3(b), only the Class Identifier and Class #B has to be retrained. This significantly allows a lot of time and resources to be saved especially when the whole dataset scales up to multiple times from the original size. Performing Transfer Learning on a specific class instead of the whole large dataset also increases the accuracy of the model since the knowledge gained from the previous training is more specific to the class.

## IV. IMPLEMENTATION OF GOOGLE CLOUD PLATFORM

AUREL makes full use of Google's Cloud Platform services to enable both online and offline operation using deep learning neural network models. While connected to the Internet, Cloud Vision API is used for image labelling as shown in Figure 4(a). Cloud Vision API is trained on a large dataset of images

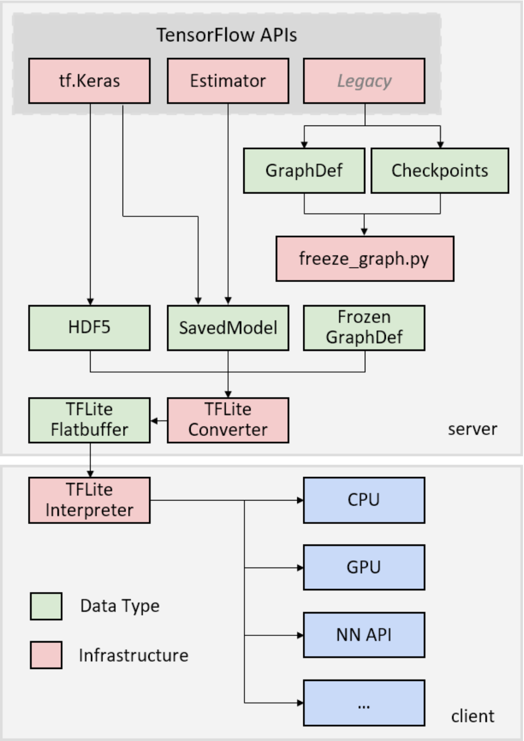


Figure 5. TensorFlow Lite conversion process

and can provide 10,000+ labels in many categories while the mobile application is connected to the cloud. It quickly classifies images into thousands of categories and improves over time as new Machine Learning concepts are introduced. In situations where there is no Internet connection, the on-device API is still able to provide 400+ labels that cover the most commonly found concepts in images. Cloud Vision API serves as the first stage of the multistage inferencing approach.

The mobile application, AUREL aims to enable users to perform on-device image classification in offline situations. Therefore, models must be able to run quickly with high accuracy in a resource- constrained environment making use of limited computation, power and space (Lane *et al.*, 2017). The selected MobileNets model is trained using transfer learning on the ImageNet Large Visual Recognition Challenge 2012 Dataset (Russakovsky *et al.*, 2017). The MobileNets model contains networks with millions of parameters that can differentiate multiple classes. By using these parameters as inputs to the final classification layer, a model can be fine-tuned accurately. Tensorboard is used to monitor and inspect the training process as it takes place. As the training progresses, a series of output steps showing the training and validation

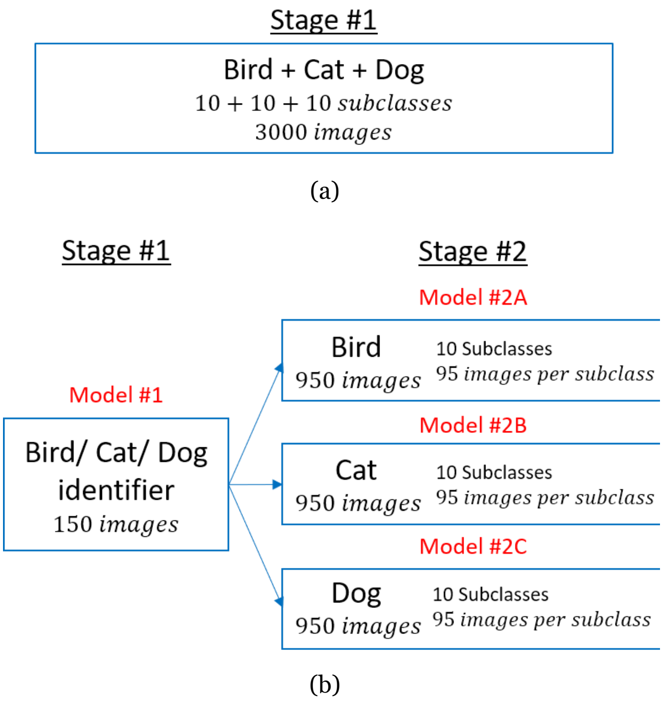


Figure 6. Experiment #1 Setup of (a) All-in-one Network, (b) Multistage Inferencing Approach

accuracy is displayed on Tensorboard. After the training is finished, TensorFlow Lite Optimizing Converter is used to optimize the trained TensorFlow model. TensorFlow Lite is TensorFlow's lightweight solution for mobile applications and embedded devices by enabling on-device machine learning inferences with low latency and a small binary size. The converter prunes unused graph-nodes and improves performance by joining operations into more efficient composite operations. While optimizing TensorFlow protobuf files which contains graph definitions and weights of the model, TensorFlow Lite uses a different serialization format which is FlatBuffers. FlatBuffers can be memory-mapped and used directly from disk without having to be loaded and parsed which allows faster startup times.

Figure 5 depicts the TensorFlow Lite conversion process. ML Kit for Firebase enables the application of Vision API onto the mobile application as well as hosting custom TensorFlow Lite models. There are many benefits of hosting TensorFlow Lite models on the cloud instead of on-device. Model versions can be easily managed and updated. This mechanism will significantly reduce the file size of the AUREL application. The image classification models will only be downloaded onto AUREL only if required. While online, these TensorFlow Lite models are used alongside Cloud Vision API, implementing the multistage inferencing approach on object classification. These models are also automatically downloaded so that on-device labelling can be done in offline situations.

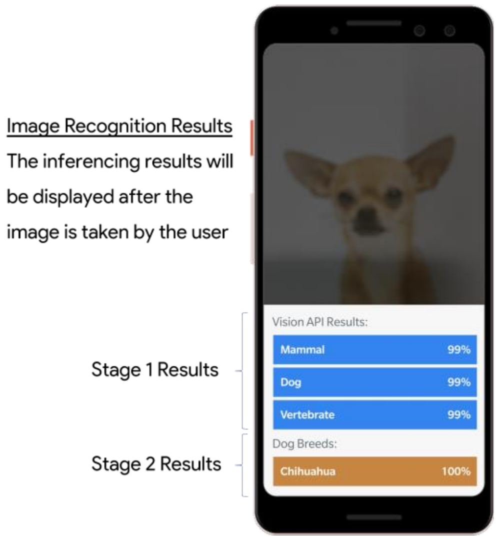


Figure 7. Object classification results displayed in AUREL

Augmented Reality is a technique of showing virtual objects on real-world images which is enabled by ARCore. A Sceneform Asset file (.sfb) is used by the mobile applications to create a renderable object to be displayed in Augmented Reality. A repository of these Sceneform Asset files is stored on Google Cloud Storage and each asset has its unique directory. Once requested by the user, the 3D model of the object will be downloaded from an online repository onto the device. The downloaded 3D models can be accessed by the user even in offline situations. Once the 3D model of the selected object is downloaded or accessed from the mobile device, ARCore and Sceneform are used to project the markerless Augmented Reality object as seen in Figure 8. The user can interact with the object and have a much more immersive learning experience.

## V. RESULTS AND DISCUSSION

All experiments were executed on desktop computer with processor Intel Xeon CPU E5-2620 v3, Nvidia GeForce GTX Titan X and 128GB DDR3 RAM @ 2133 Mhz. All experiments were trained using MobileNets for 5 epochs due to the use of pre-trained model from ILSVRC. As demonstrated in Figure 6, the dataset for the Experiment #1 consists of 10 dog subclasses, 10 cat subclasses and 10 bird subclasses, totaling to 30 subclasses. Each subclass contains 100 images for training and 20 images for inferencing. By using the All-in-one network, all 30 subclasses are trained together, producing only a single model as seen in Figure 6(a). As for the multistage inferencing approach, for the first stage, five random images from each of the 30 subclasses are first selected. The images are grouped into 3 main classes, which are bird, cat and dog to form the first stage dataset. The model produced is known as Model #1. As for the remaining 95 images in each of the 10 dog subclasses, training is done to produce Model #2A. This step is repeated for the cat and bird classes to produce Model #2B and #2C respectively. Once all training is done, the evaluation process takes place and the recognition rate is calculated using the equation below:

Table 2. Recognition Rate for Experiment #1 and Experiment #2

| Experiments | Recognition Rate | | | |
|---|---|---|---|---|
| | Experiment #1 | | Experiment #2 | |
| | Stage #1 | Stage #2 | Stage #1 | Stage #2 |
| All-in-one Network | 88.5% | - | 87% | - |
| Multistage Inferencing Approach | 98.5% | 89.67% | 98.73% | 90.68% |

Table 3. Total time taken for training 1 epoch

| Approach | All-in-one Network | Multistage Inferencing Approach |
|---|---|---|
| Stage #1 | 13 min 5 sec | 39 sec |
| Stage #2A | - | 4 min 2 sec |
| Stage #2B | - | 4 min 13 sec |
| Stage #2C | - | 4 min 9 sec |
| Total Time: | 13 min 5 sec | 13 min 3 sec |

$$\text{Recognition rate} = \frac{\text{Correctly labelled images}}{Total\ images} \times 100\% \qquad (3)$$

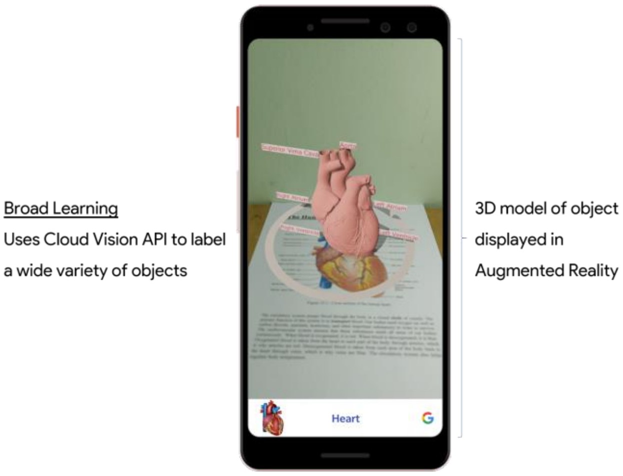
For the All-in-one network, inferencing is done directly on 20 images for each subclass, totaling to 600 images. 531/600 images are correctly labelled, resulting in a recognition rate of 88.5%. As for the multistage inferencing approach, inferencing is done twice in this case. For the first stage, Model #1 is used to identify the general classes between bird, cat and dog with a result of 98.67% accuracy. For the second stage, depending on the results of the first stage, if the result is dog, cat or bird, Model #2A, #2B or #2C will be used respectively. For this stage, a recognition rate of 89.67% is obtained. From the results obtained, there were minor differences observed between All-in-one network and multistage inferencing approach at this small scale of dataset as shown in Table 2.

Table 4. Sample images which are corrected labelled using multistage inferencing approach but incorrectly labelled using all-in-one network

| Sample Images | Wilson's Warbler | Ragdoll | Beagle |
|---|---|---|---|
| All-in-one network | Western Meadowlark | Birman | English Foxhound |
| Multistage Inferencing Approach | Wilson's Warbler | Ragdoll | Beagle |

Experiment #2 is repeated with the extended dataset containing 10 cat subclasses, 50 bird subclasses and 50 dog subclasses. Each subclass contains 100 images for training and 20 images for inferencing. For the all-in-one network, inferencing is done directly on 20 images for each subclass, totaling to 2200 images. A recognition rate of 87% is obtained. Moving on to the multistage inferencing approach, at the first stage, a recognition rate of 98.73% is obtained. As for the second stage, a recognition rate of 90.68% is obtained. For this experiment, by scaling up the dataset upwards to approximately 4 times the original size, a 3.68% increase in recognition rate is successfully obtained as shown in Table 2.

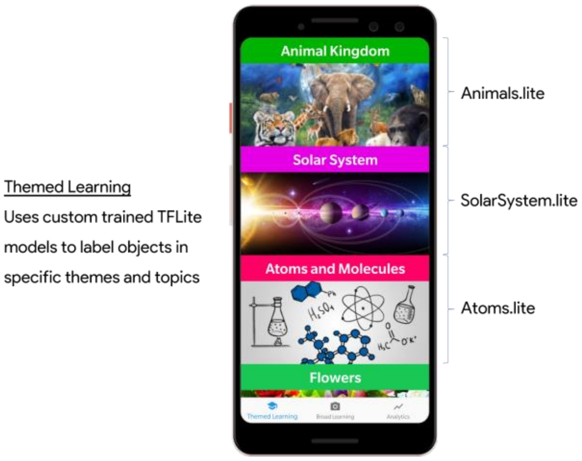
In Table 3, the total training time for both approaches are observed to be similar since the total number of images used are the same. In AUREL, the object classification results in both stages of the Multistage Inferencing Approach will be displayed if the confidence exceeds the preset threshold of 70%. As shown in Figure 7, the list of Cloud Vision API results, which is the first stage is displayed on top of the page list. Since "Dog" is one of the results in the first stage, Model #2C will automatically be used in the second stage. The result of the second stage is displayed at the bottom. Sample images which are corrected labelled using multistage inferencing approach but incorrectly labelled using all-in-one network are shown in Table 4.



Broad Learning
Uses Cloud Vision API to label a wide variety of objects

3D model of object displayed in Augmented Reality

Figure 8. Broad learning Page



Themed Learning
Uses custom trained TFLite models to label objects in specific themes and topics

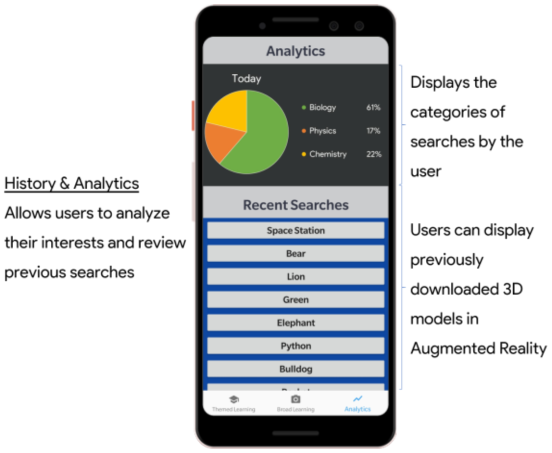
Animals.lite

SolarSystem.lite

Atoms.lite

Figure 9. Themed Learning Page

In the design of AUREL, there are three main pages on the mobile application, which are Broad Learning, Themed Learning and History and analytics. In Broad Learning, the user can take a picture of any object that the user is interested to learn more about. If the device is connected to the Internet, the application uses multistage inferencing approach to detect the object in the picture. This functionality allows the user to search for objects even if they don't know what the name of the object is. A list of the top results and the corresponding confidence is displayed on the screen of the device. Once the user selects one of the results, its 3D model will be downloaded from the 3D model repository on Google Cloud Storage onto the device. The user is then able to point their device onto a flat surface and display the 3D model of the object in Augmented Reality using ARCore. The user can interact with the object such as rotating and scaling the object to fully visualize the object in real life.



History & Analytics
Allows users to analyze their interests and review previous searches

Displays the categories of searches by the user

Users can display previously downloaded 3D models in Augmented Reality

Figure 10: History & Analytics Page

Themed Learning is used for situations whereby the user is interested to explore specific topics such as "Atoms and Molecules", "Animal Kingdom" and "The Galaxy". Each of these topics will use their respective TensorFlow Lite models to identify objects for the user. These TensorFlow Lite models are hosted on MLKit for Firebase and will only be downloaded onto the application when required hence decreasing the file size of the application. This mechanism also allows the Image Labelling models to be constantly updated on the cloud without the user having to reinstall the mobile application each time the models are updated. The search history of the user is also properly documented into categories such as Physics, Biology, Astronomy and many more. These data are displayed statistically so that the user can get a better understanding of their interests. Another feature is that the user can send feedback on wrongly identified objects so that the machine learning models can be corrected.

## VI. CONCLUSION

Conventional methods of using All-in-one network when training large datasets is tedious and challenging. Problems such as algorithm crashes and out-of-memory error are likely to occur when training on large datasets. By using the proposed multistage inferencing approach, large datasets can be divided into multiple smaller datasets, arranged hierarchically and form a pipeline for object classification. This approach of object classification was demonstrated in the experiments with 90.68% of recognition rate as compared to all-in-one network. As the dataset sizes become larger, the training process of the multistage inferencing approach could be reduced by focusing on the additional subclasses and the main object identifiers. AUREL, a mobile application platform is developed to identify objects for STEM education learning and display the detected objects in Augmented Reality. For future works, the application of transfer learning onto subclasses in the multistage inferencing approach can be studied to further improve the accuracy.

## VII. REFERENCES

Ang, I.J.X. & Lim, K.H. 2019, "Enhancing Stem Education Using Augmented Reality and Machine Learning", in *2019 7th Int. Conf. on Smart Computing and Communications (ICSCC) pp. 1-5*.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. & Fei-Fei, L. 2009, "Imagenet: A Large-Scale Hierarchical Image Database", in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, Fl, pp. 248-255.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. & Adam, H. 2017, "Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications".

Krizhevsky, A., Sutskever, I. & Hinton, G.E. 2012, "Imagenet Classification with Deep Convolutional Neural Networks", *Advances in Neural Information Processing Systems*, pp. 1097-1105.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T. & Ferrari, V. 2018, "The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale".

Lane, N.D., Bhattacharya, S., Mathur, A., Georgiev, P., Forlivesi, C. & Kawsar, F. 2017, "Squeezing Deep Learning Into Mobile and Embedded Devices", *IEEE Pervasive Computing*, vol. 16, no. 3, pp. 82-88.

Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. 1998, "Gradient-Based Learning Applied to Document Recognition," in *Proceedings of the IEEE, vol. 86*, no. 11, pp. 2278-2324.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C.L. 2014, "Microsoft Coco: Common Objects in Context", *European Conference on Computer Vision*, pp. 740-755.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. & Berg, A.C. 2015, "Imagenet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision*, vol. 115, no. 3, pp.211-252.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. 2015, "Going Deeper With Convolutions", in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. 2016, "Rethinking The Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, Nv.

Tucker, B. 2012, "The Flipped Classroom", *Education Next*, vol. 12, no. 1, pp.82-83.

Wang, J., Cao, B., Yu, P., Sun, L., Bao, W. & Zhu, X. 2018, "Deep Learning Towards Mobile Applications", in *2018 IEEE 38th International Conference On Distributed Computing Systems (ICDCS)*, Vienna, pp. 1385-1393.

Xie, Y., Fang, M. & Shauman, K. 2015, "Stem Education", in *Annual Review of Sociology*, vol. 41, pp. 331-357.