Curry AI: A Sophisticated AI Application for Nutritional Calculator

Tuning Multi-class Classification for Large Number of Distinct Classes

In my previous paper on CurryAI, I described a computer vision aided Indian Food Nutrition Calculator (https://analyticsindiamag.com/how-i-created-curryai-a-computer-vision-aided-indian-food-nutrition-calculator/).

The crux of this work is using deep learning, and particularly transfer learning, in image recognition to distinguish Indian Food dishes from their photographs. This work can further be used to calculate calorie and nutritional content of a particular dish by using the USDA Food Central database that maps a dish to its components and nutritional content.

My earlier work used a training dataset of 31 distinct dishes (IndianFood31) to train and test a multi-class model. The model was able to identify and distinguish fairly well between 31 classes.

As 31 dishes are too few to represent the diversity of Indian cuisine, I aimed to increase the number of dishes the calculator could identify. In an ideal world, this algorithm would be able to distinguish between thousands of food dishes with perfect precision. However, image recognition algorithms perform well when number of classes is small (cat vs dog etc.)

Multi-class classification becomes exponentially complex when it needs to distinguish between larger number of classes.

Therefore one needs to progress to the ultimate goal in phases. In this second phase of CurryAI, I doubled the dataset from 31 to 62 dishes. This dataset is called as IndianFood62 (See Appendix).

I wanted to observe how the deep neural network's performance changes with doubling of the classes, understand the factors that affect its ability to distinguish between the classes and how the factors or hyperparameters can be tuned to provide improved performance. In machine learning, a **hyperparameter** is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are derived via training. (*Wikipedia*, 2022)

Deep Neural Network Architecture

I used a pre-trained deep learning neural network called MobileNetV2. This network which has been trained on image classification with hundreds of thousands of varied images of 1000+ categories. It is an established practice in image classification to use a pre-trained network rather than start from scratch. The final layer of this network is discarded, and further training is performed on the custom dataset. This results in a phenomenon called

'transfer learning' wherein the learning of pre-trained network is further augmented and transferred to the specific problem at hand (in this case food image recognition).

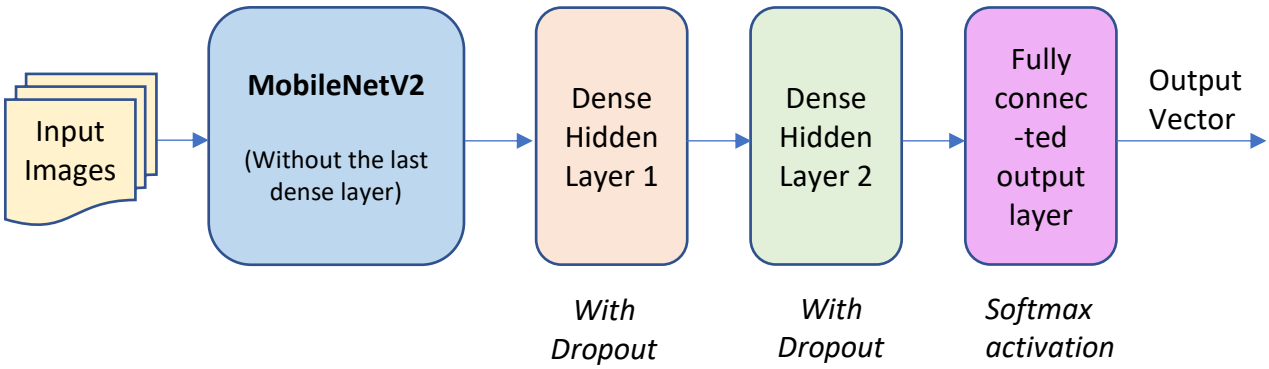

Figure 1: Architecture of CurryAI Deep Neural

Hyperparameter: Units or Number of Neurons in Dense Hidden Layer

One of the factors that affects the deep neural networks learning ability is the units or number of neurons or nodes in the dense hidden layers. A dense layer is deeply connected layer from its preceding layer which works for changing the dimension of the output by performing matrix vector multiplication. Results from every neuron of the preceding layer go to every single neuron of the dense layer. *(Verma, 2021)*

Units is a basic and necessary hyperparameter of the Keras dense layer which defines the size of the output from the dense layer. It represents the dimensionality of the output vector. Adding more units to the dense layer increases the complexity of the model as well as the capacity to learn. However there is a trade-off between ability of the model to learn and over-fit. An overly complex model may learn the noise in the data as well, in other words 'over-fit', and thereby perform poorly on unseen data, defeating the purpose of the model.

For our experiment, we start with dense hidden layers of 31 neurons each with a 20% dropout and an output layer of 62 neurons. We then increase the number of neurons to 62 in the second run and then 124 in the 3rd and final run.

We measure the performance of the algorithm using Mean Average Precision, Recall and F1-score which is a popular way to understand how an object detector algorithm is effective across all the multiple image classes *(Hui, 2018).* Precision indicates how many true positives are found when the algorithm makes a prediction, and Recall indicates how many instances of the ground truth set it is able to identify. F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean.

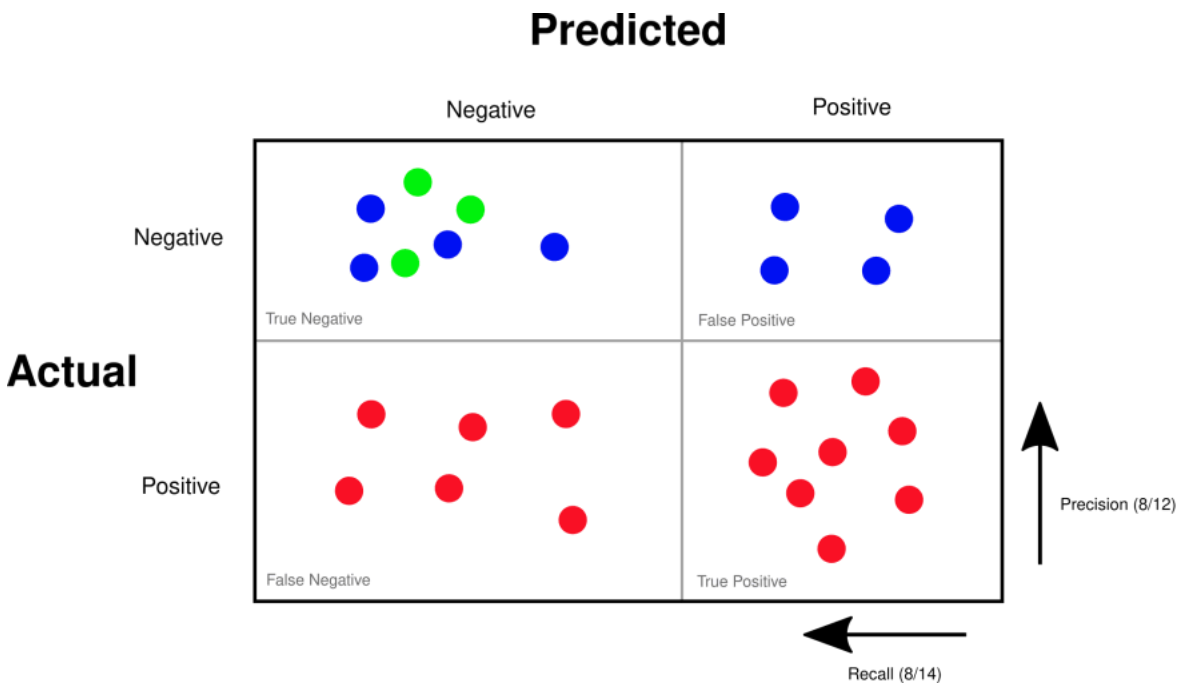

Figure 2: Model Performance Measurement – Precision & Recall (Company, 2020)

We find that F1-score of the model increases significantly when units are increased from 31 to 62 and further increases slightly from 124 but at an expense of increased run-time. Moreover, while increases when units are increased from 62 to 124, Weighted Mean Average Precision and Recall remains about the same. This means that increase in complexity and run-time is not justified when units are increased from 62 to 124. The probable reason is that the model starts to overfit and learn the noise and hence increased capacity of network is not really helpful any longer.

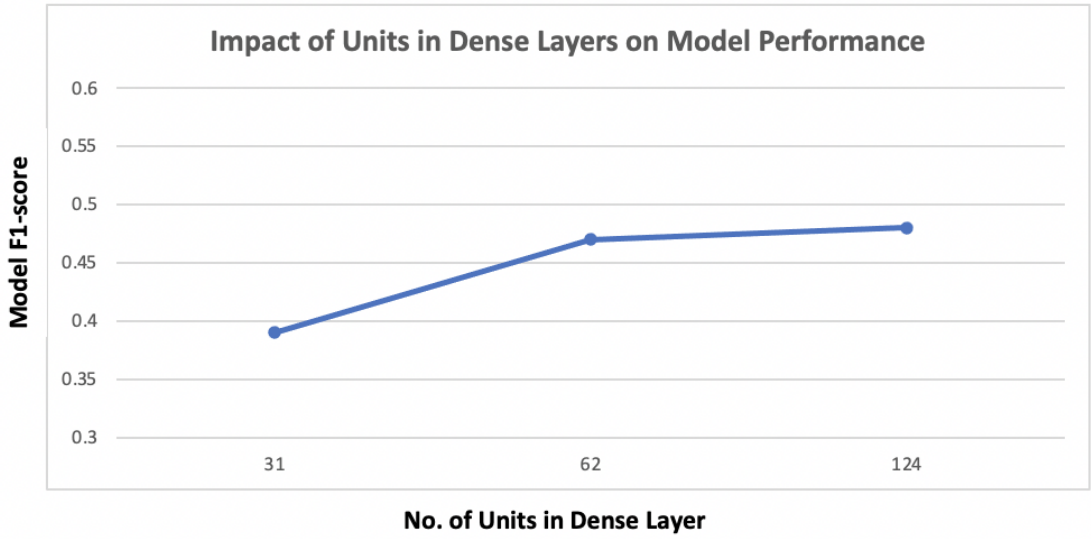

Figure 3: Impact of Units in Dense Layers on Model Performance

Hyperparameter: Dropout

Deep neural nets with a large number of parameters are very powerful machine learning systems. However, overfitting is a serious problem in such networks.

Large neural networks also have long and costly run-times, therefore using ensembling or parallel runs to combine predictions of multiple networks is an expensive process. Instead an alternative technique to address this problem is 'dropout'. Dropout is a type of regularization method, that is, it helps the model to generalize better over the data and reduce the variance of the model.

The key idea is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much and significantly reduces overfitting. *(Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014)*

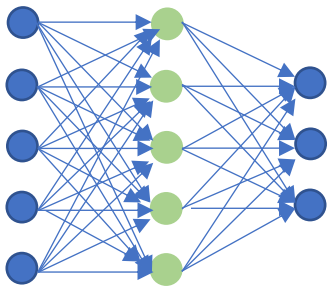

Figure 4: Fully Connected Deep Neural Network

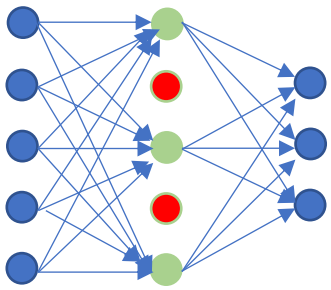

Figure 5: Deep Neural Network with Dropout

For our experiment, we start with dropout of 20% within the dense hidden layers and then increase it to 25% and then 30%. We find that an increase in dropout from 20% to 25% improves the performance, due to reduction in over-fitting. However, further increase in dropout from 25% to 30% dramatically reduces the performance.

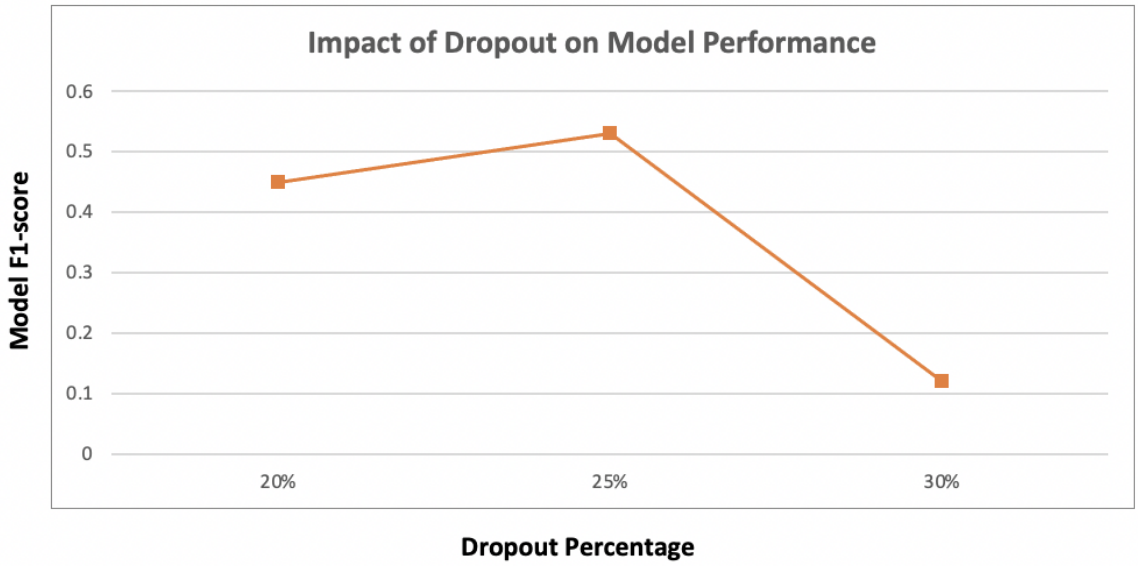

Figure 6: Impact of Dropout on Model Performance

Conclusion

The deep learning model and computer vision calculator implemented for CurryAI is now expandable to all food and is able to identify to dishes from their photographs.

As we double the number of Indian Food Dishes (from 31 to 62), the CurryAI deep neural network finds the task challenging and thus more training datasets are needed to continue to fine tune the models.

To improve the neural network performance, we tune 2 key hyperparameters – units and dropout. This leads to an improved performance of the model with F1-score of 0.53 and Mean Average Precision of 0.55 and Mean Average Recall of 0.58. The model is able to distinguish well between distinct dishes but performs poorly particularly on gravy items such as butter chicken, chicken gravy and paneer gravy, which are very similar looking.

Any further increase in performance in the model will be achieved by further training data augmentation rather than further tuning of hyperparameters.

With the foundation of the deep learning model set, we will now move to developing a mobile application to get this applicable in the day to day usage for millions of Indians, tuned to the food and dishes of the country.

Appendix

IndianFoodList62

Sr. No.	Indian Dish Name

1	Aloo Mash
2	Avial
3	Baati
4	Barfi
5	Bonda
6	Butter Chicken
7	Chaat
8	Chakli
9	Chana Gravy
10	Chevda
11	Chicken Dry
12	Chicken Gravy
13	Dahi vada
14	Dal
15	Dosa
16	Egg Gravy
17	Gajar Halwa
18	Gobhi Sabzi Dry
19	Gulab Jamun
20	Halwa
21	Idli
22	Imarti
23	Jalebi
24	Kachori
25	Khaman dhokla
26	Khandvi
27	Kheer
28	Khichdi

29	Kulfi falooda
30	Laddu
31	Malai Kofta
32	Malpua
33	Modak
34	Momo
35	Naan
36	Noodles
37	Paneer Gravy
38	Pani Puri
39	Paniyaram
40	Papad
41	Paratha
42	Payasam
43	Peda
44	Poha
45	Raita
46	Rajma
47	Rasam
48	Rasgulla
49	Rongi
50	Roti
51	Paneer Palak
52	Sambar
53	Samosa
54	Sevai
55	Shankarpali
56	Sohan Papdi

57	Aloo Tikki
58	Undhiyu
59	Upma
60	Uttapam
61	Vada
62	Vada pav

Bibliography

Wikipedia. (2022, July 27). *Hyperparameter (machine learning)*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning)

Verma, Y. (2021, September 19). *A Complete Understanding of Dense Layers in Neural Networks*. Retrieved from Analytics India Magazine: https://analyticsindiamag.com/a-complete-understanding-of-dense-layers-in-neural-networks/

Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov. (2014, June). *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. Retrieved from Journal of Machine Learning Research: https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf

Hui, J. (2018, March 7). *mAP (mean Average Precision) for Object Detection*. Retrieved from Medium: https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173

Company, P. M. (2020, July 29). *When accuracy is not enough...* Retrieved from DEV Community: https://dev.to/polmonroig/when-accuracy-is-not-enough-2hej

About the Author

Shubh Samtani is a XII grade student at The International School Bangalore (TISB). He is the creator of Curry AI, a sophisticated machine learning based application that uses deep image recognition to assess the nutritional value of food by taking a picture of the food.

Shubh is the global winner of MARRS International Spelling Bee Contest which had participation from over 250,000 children. Shubh is among the top programmers globally on HackerRank platform, which has over 14 million active programmers. He has been one of the youngest to achieve a ranking below 10,000 at an age of 14 years globally on the platform.

Shubh volunteers as a teacher at 0Gravity, a global movement to create awareness, structured training programs and communities for computer education for kids 10-15 years

old. He has been primarily responsible for porting all coursework of 0Gravity online for thousands of students, and designing and delivering the course – 'Solving Analytical Problems Using Python'.

For more than last couple of years, 2020-2022, Shubh has conducted and published ground breaking research, under the guidance of Dr. Suresh Bhagavatula from IIM Bangalore, on India's National Education Policy (NEP) and assessing the impact on the Right to Education (RTE) among 400 million children in India during the COVID-19 pandemic.

Shubh has completed the niche and highly selective global program for high school entrepreneurs offered by LaunchX (MIT Program) in 2021.