1. For this research, a publicly available dataset provided in Kaggle for Microsoft Malware Classification Challenge (BIG 2015) [19] is utilized. This database consists of malware programs belonging to 9 different categories. Separate training and testing datasets have been provided in BIG 2015 challenge, however, labels for the testing dataset is not publicly available. Hence, we solely utilize the training dataset in BIG 2015 in Order to estimate our performance. Figure 1 shows the distribution of the training dataset of malware programs. Table 1 displays their corresponding IDs and malware category. For each malware program, an assembly and a bytes file has been provided. Only the bytes file is used to visualize malware programs as images as presented in research work. The dataset is randomly split into groups of 72%, 8% and 20% for training, validation and testing purposes respectively. Figure 3 and Table 2 shows the distribution of training, validation and testing datasets respectively.

2.In this paper, we use the malware dataset provided, courtesy of Microsoft, for the Microsoft Malware Classiﬁcation Challenge (BIG 2015) [24], [29]. The full suite BIG 2015 database contains 10868 labeled malware programs of 9 different categories. For each program, a disassembled (low-level language) ‘.asm’ ﬁle and the corresponding compiled (CPU Machine Code) ‘.bytes’ ﬁle are provided. In our experiment, we opt not to include the category ‘Simda’ (labeled class 5) because it only contains 42 samples and hence is deemed inadequate for the purpose of training an architecture like that of a deep learning (which generally requires a much higher number of training samples for a given class) [21]–[23]. That is to say, in our experiment we make use of 10826 malware programs of 8 different classes to train and test our system. More speciﬁcally, out of the 10826 malware programs, we make use of 75% of the dataset for training and the remaining 25% for testing. It is worth emphasizing that, to rigorously evaluate our approach, only the 75% of the data set aside for training is used to tune all the parameters of our deep learning based system. This essentially constitutes a hold-out validation analysis. Note that we made sure that for a given class we do have 75% of the programs/samples of that class within training set and 25% of data of that same class within the testing set (i.e., evenly distributed within training and testing sets). Moreover, for each class, this was done in a completely randomized fashion. Figure 1 shows the distribution of the malware programs used to train and test our system.

3. The malware dataset is almost half a terabyte when uncompressed. It consists of a set of known malware ﬁles representing a mix of 9 diﬀerent families. Each malware ﬁle has an identiﬁer, a 20 character hash value uniquely identifying the ﬁle, and a class label, which is an integer representing one of the 9 family names to which the malware may belong (See Table 1). For each ﬁle, the raw data contains the hexadecimal representation of the ﬁle’s binary content, without the header (to ensure sterility). The dataset also includes a metadata manifest, which is a log containing various metadata information extracted from the binary, such as function calls, strings, etc. This was generated using the IDA disassembler tool. The original question posed to participants was to classify malware to one of the 9 classes. The dataset can be downloaded from the competition website.

4. We evaluated the experiment using a publicly available Microsoft dataset, which was made public in 2015, with a size of nearly 0.5 TB. Each malware ﬁle has an identiﬁer, a hash value, and a class label. Each of the raw data has a hexadecimal representation of the binary content. The title is not included to ensure sterility. The dataset has two different types of data: one is byte format, and the other is asm format. As the dataset has already been processed, it can be used directly. We converted the bytes in the binary ﬁle sample into images. The dataset contains 10,868 malware samples from nine families. The distribution of malware samples in the Microsoft dataset is shown in Table.

5. In 2015, Microsoft hosted a Kaggle competition for malware classiﬁcation [8]. In this challenge, Microsoft released a huge dataset (almost half a terabyte when uncompressed) consisting of 21,741 malware samples. This dataset is divided in two parts, 10,868 samples for training and the other 10,873 samples for testing. Each malware sample belongs to one of 9 different malware families. Like the Malimg dataset, the distribution of malware samples over classes in the training data is not uniform and the number of malware samples of some families signiﬁcantly outnumbers the samples of other families. There are two ﬁles that represent each malware sample, .bytes ﬁle that contains the raw hexadecimal representation of the ﬁle's binary content with the executable headers removed and .asm ﬁle that contains the disassembled code extracted by the IDA disassembler tool. In our experiments, we only use the .bytes ﬁles to generate the malware images. We experiment with two different settings involving this dataset: 1) Setting-A: following previous work [19], [20], the original training set (10,868 malware samples) is class-wise randomly divided into two subsets where the ﬁrst set consists of 90% malware ﬁles (9,776) and the rest 10%(1092) is used for testing; 2) Setting-B: we follow the original train-test split which is provided by Microsoft (i.e. 10,868 samples for training and the other 10,873 samples for testing).

6.

7. Microsoft provided a dataset composed of21741 samples for theBig Data Innovators Gathering (BIG2015) Anti-Malware Prediction Challenge, 10868 for training and 10873 for testing. Every program in the dataset has a ﬁle containing the hexadecimal representation of the malware’s binary content and its corresponding assembly ﬁle. However, only the label for the samples belonging to the training dataset is provided. Table 5 shows the distribution of malware programs present in the training dataset.

8. In 2015, Microsoft held a malware classification challenge, using a malware dataset hosted on Kaggle. BIG 2015 is a 500-gigabyte dataset of malware files released by Microsoft in 2015. This dataset consists of malware samples from nine families: Rammit, Lollipop, Kelihos\_ver3, Vundo, Simda, Tracur, Kelihos\_ver1, Obfuscator, and Gatak. These families each fall into one of six different varieties of malware: worm, adware, backdoor, trojan, trojan downloader and obfuscated malware. Due to the obvious downsides of releasing large quantities of dangerous executable files, the binaries are all dissassembles using the IDA interactive disasembler. For each specific sample, there is an IDA output file, consisting of the full assembly code and raw byte data, as well as a fle consisting only of the bytes and line numbers. Microsoft hosted a competition among data scientists and security professionals to create a program to classify these malware families to identify future malware. All files in the dataset are known files, meaning they have been determined to contain malicious code [2] and there are no clean, non malware samples or categories of files. This means that a DCNN trained using samples from this dataset will only be trained to clasify malware families, and will not be able to differentiate malware from non-malware.

9.

In this paper, we use the malware dataset provided by Kaggle for Microsoft malware Classiﬁcation Challenge (BIG 2015) [21]. BIG 2015 database comprises of two separate sets: a training dataset and a testing dataset. Every program present in BIG 2015 dataset has a corresponding assembly ﬁle and ‘.bytes’ ﬁle depicting the characteristics of the particular program. The training dataset comprises of 10868 classiﬁed malware programs into 9 publicly known different categories. Table I shows the distribution of various classiﬁed (see Class ID column) categories present in the training dataset. The testing dataset contains 10873 programs. However, the labels of the programs in the testing set are not publicly available.In this paper, we focus only on the provided training dataset for our evaluation purposes.