Using Computer Vision to Study the Microstructure of Heusler Alloys

Shambhu KC, Tecia Grier

PH-582 Machine Learning

Final Project Report: Team L

Abstract

The microstructure of a material contains extensive details about the material such as phases, grain structures, elemental distribution etc. In this work, we use computer vision to recognize the microstructure images of the alloys to classify Heusler alloys into single-phase or multiphase classes. In general, a multi-phase microstructure exhibits different contrasts pertaining to different phases. Such a behavior, however, does not always mean that different contrast regions have different elemental composition. A human expert with many years of related experience or the Energy Dispersive X-ray Spectroscopy (EDS) is then required to correctly characterize the microstructure. As an alternative approach, deep learning networks can be trained so that they learn from the patterns in the microstructure and can make correct predictions about the phase distributions. This deep learning approach will not only reduce the need for a human expert or the EDS, but it will also expedite the process of microstructure characterization thereby leading to an accelerated material discovery.

1. Introduction

Heusler alloys are intermetallic compounds of transition metal and main group elements. In general, these alloys are ternary with the general formula X_2YZ (full Heusler) or XYZ (half Heusler), but there are also other variants, such as quaternary (XX'YZ), binary (X_3Z), and substitutional ($X_{1-a}X'_aYZ$). The X and Y are generally transition metal elements whereas Z represents main group elements. A special property of Heusler alloys is that they are tunable, meaning a material of desired properties can be easily tailored ranging from half-metallic ferromagnets[1], super-conductivity[2], topological insulator[3], non-collinear magnetism[4], thermoelectric effects[5]. These wide range of properties make them suitable candidates in many practical applications, such as spintronics[6], solar cells[7], thermoelectric devices[8], and many others.

As Heusler alloys are made by mixing different elements, it is necessary to confirm if solid solution is formed after the mixing – the formation of solid solution is primarily determined by the difference in the atomic number and in the electronegativity of the constituent elements[9]. In other words, data analysis is necessary to determine if the alloy is stable in a single-phase compound or it decomposes into multiple phases. The formation of a single-phase compound is necessary particularly for accessing structural, magnetic, and electronic properties. If there are multiple phases, the desired properties will be compromised, and the ability to determine the type behavior that correlates with each phase would be difficult.

The study of microstructure images, micrographs, is a fast and cost-efficient technique used to investigate the phase-stability of materials. A microstructure is material structure at the

micrometer to nanometer scale that contains information regarding the grain size, different phase size, and distribution, inclusions, and impurities of the material. Microstructure is also useful in studying the role of different operations such as heat treatment, mechanical treatment. Hence by observing the microstructure, useful information regarding the material can be extracted from the micrograph using limited resources.

In microstructure, the different phase regions appear to have different contrasts. This is due to selective reaction rate of etchant. An etchant is a reagent used to reveal the microstructure on different chemical species. Therefore, sufficient predictions of the phase stability of a material can be concluded by observing the contrast pattern. However, in some cases, the grains of the same chemical compositions are attacked differently by the etchant which leads to different contrast between them. Therefore, a human expert with many years of experience is generally required to correctly interpret and characterize the microstructure images. The heavy reliance on the researcher's experience can, however, introduce bias and potential error in the process of microstructure recognition. To address this problem, there has been growing interest towards building automated micrograph recognition tools based on machine learning (ML) and deep learning (DL) that require no prior knowledge of microstructure features.

Fueled by the exponential growth in computation power, the availability of large dataset and the improvements in the algorithms, the ML has proven to reach or even surpass human abilities in numerous tasks (such as playing the game of Go[10], image classification[11], predicting weather[12] etc). Just like in other fields, the ML has become one of the most useful tools in the field of materials science by expediting the material discovery process. This time and cost efficient method also promotes innovative ideas to discover revolutionary functional materials. A few examples of successful implementation of ML in Materials Science are prediction of the stability of material[13, 14, 15], estimation of numerous material properties[16, 17, 18], and speeding up the first principles based calculations[19].

In addition to aforementioned tasks, the ML has also been used for several microstructure image recognition tasks. For instance, DeCost et. al.,[20] utilized support vector machine (SVM) with χ^2 kernel to classify microstructure images into seven target classes: brass, ductile cast iron, gray cast iron, hypoeutectoid steel, malleable cast iron, superalloy, and annealing twins. By using a dataset of 105 images, they were able to classify the images with 80 % accuracy. Similarly, Chowdhury et.al.,[21] employed different ML algorithms to classify the dendritic microstructure from non-dendritic counterparts. In their work, a pre-trained convolutional neural network (CNN) achieved the best accuracy (~92 %). More recently, Chen et. al.,[22] utilized Mask R-CNN to identify features at pixel level based on instance segmentation. However, research involving the classification of microstructure in terms of phase-stability (i.e., classifying whether it is single-phase or multi-phase) has not been reported in the literature. Using ML models limits the use of other characterization techniques leading to less manual efforts and lower cost. Further research involves implementing instance segmentation with regression tasks so that different phase fractions can be automatically estimated. This spurred us to embark on building ML models that help characterize the microstructure images.

2. Proposed Methodology

The goal here is to identify the single-phase or multiphase microstructure images based on the contrast patterns. As mentioned, the contrast difference is mainly due to the preferential reaction rate of etchant, which could be due to having different chemical compositions or different crystallographic orientation. However, impurities such voids, scratches made during the polishing, cracks can sometimes appear darker in optical micrographs which can misconstrue the contrast pattern. The model should be robust enough to learn these features and classify the images correctly independent of contrast differences. The success of this task depends on the feature extraction, where the feature represents local regions of pixels that have similar interesting properties such as edges, corners, and contrast differences. The computer vision algorithm, particularly CNN, has proven to outperform any other feature extraction tools by far. In addition, CNN can perform unsupervised feature extraction, hence no prior knowledge of microsturce features is required. There, CNN will be used to build this model.

CNN is a multi-layer neural network that focuses on recognizing the patterns from pixel images; it is a high accuracy tool for image recognition and classification. Each neuron is defined by its weights, also known as filters or kernels. If all the neurons in a given layer share the same filter, then the layer outputs a feature map. This feature map highlights the areas in an image that activate the filter the most. Each neuron in a feature map takes a patch of pixels as input i.e., pixels in the receptive field, from the previous layer. This allows the network to learn low level features, such as horizontal, vertical, and diagonal edges, in the starting layers and then assemble them into high level features, such as objects, faces, at upper layers. Since all the neurons in a feature map share the same weights, once the network recognizes a pattern in a particular location, the same pattern can be recognized in any other location. In general, a convolution layer contains many features maps (also called filters), which allows the network to learn many different features. The CNN models are trained by backpropagation, in which the gradient of the loss function is back propagated down to the first layer and the neuron’s weights are updated based on the gradient descent principle. At the start of the training process, the neuron's weights are initialized randomly but as the training progresses these weights are updated in such a way that the predicted output is moving close to the actual weights.

A typical CNN architecture, Figure 1, consists of a couple of convolution layers, each with an activation function (like ReLU, ELU, Tanh). The convolution layers are followed by a pooling layer. The pooling layer is similar to the convolution layer in the sense that each neuron in a pooling layer is connected to a certain number of neurons from the previous layer, which lie in its receptive field. The neurons in the pooling layer, however, do not have weights associated with them; the pooling layer aggregates the inputs by using an aggregation function, such as maximum, average. The pooling layers helps in reducing the computation load, memory usage, the number of parameters and also introduces rotational invariance in the learned features. The combination of convolution and pooling layer is repeated several times (the optimum number of repetitions depends on the type of problem and can be treated as a tunable hyperparameter) to make the model deeper. The deeper model generates more feature maps, therefore the model can learn complicated features in the images. It is also useful to add the batch normalization (BN) and a dropout layer in the CNN architecture. The BN layer helps in avoiding vanishing/exploding gradient problems during the training, by zero centering and then normalizing the inputs. The BN layer is typically added after the activation but in some cases it

is also implemented before the activation function. The dropout layer, on the other hand, acts like a regularizer and helps in avoiding the overfitting issue. The main idea behind the dropout layer is to drop a random number of neurons (the number is defined by dropout rate) during the training. This means outputs of those neurons will be excluded during a training step but they can be active for the next training step. If the model is overfitting, the dropout rate should be increased, whereas the rate can be decreased if the model is underfitting. Finally, at the top of the CNN, a few fully connected (FC) layers are added followed by a final output layer that outputs the predicted class probabilities. The number of FC layers and the number of neurons in the layers can be treated as a tunable hyperparameter. The number of neurons in the final layer depends on the number of classes; for example for binary classification one neuron with sigmoid activation function is sufficient.

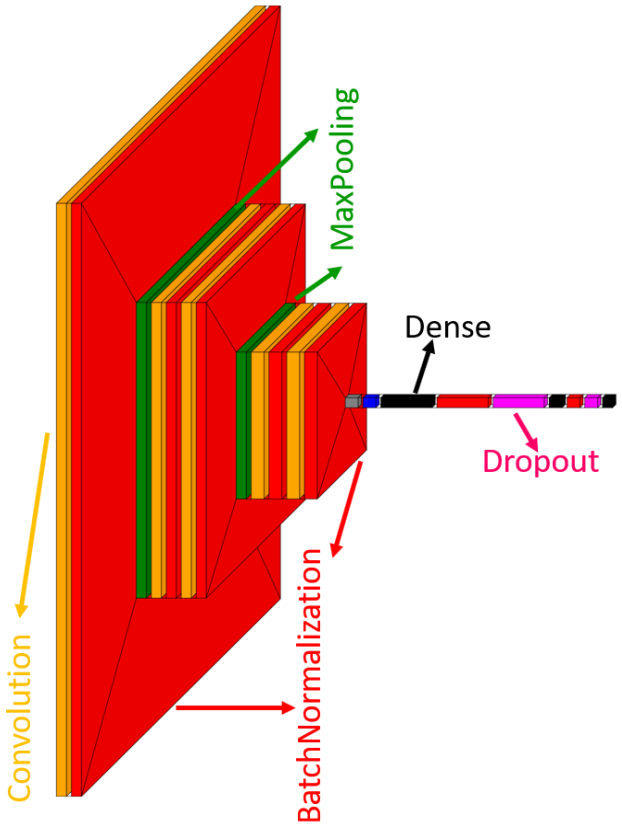

Fig 1: A typical CNN architecture

In this project, different CNN architectures will be trained to produce the best performance model. We will start with the architecture, as shown in figure 1. The optimizer, activation function, the number of filters in the convolution layers, the number of neurons in the dense layers and the learning rate with the best outcome will be tuned through gridsearch. Next, several state-of-the-art architectures such as GoogLeNet, VGG16, and ResNet will be trained. These architectures are described briefly below.

GoogLeNet architecture, which was developed by researchers at Google won the ILSVRC 2014 challenge. This architecture uses inception modules. These inception modules help the model to use parameters more efficiently resulting in a deeper and wider model with a constant computation budget but yet achieving a better performance[23]. These inception models use a combination of convolution layers with kernel sizes 1x1, 3x3, and 5x5. The different kernel sizes allows models to capture features at different scales. GoogLeNet is also known as Inception. There are different versions of Inception such as viz. v1, v2 and v3. This project will utilize the latest version of Inception, InceptionV3.

VGG16 architecture was a runner up in the ILSVRC 2014 challenge. This network was proposed by K. Simonyan and A. Zisserman from the University of Oxford[24]. This architecture can be taken as improvement over AlexNet, due to the replacement of the larger kernel size. The architecture uses repetitions of two or three convolution layers followed by a pooling layer.

ResNet, which was developed by Kaiming He et al., was the winner of ILSVRC 2015 challenge. This network uses Residual Network (hence the name ResNet) with skip connections, that is connecting input directly to the output (by skipping one or more layers). This type of connection introduces identity mapping, and helps solve the degradation problem of deeper networks by increasing the depth of a network, the model accuracy saturates and then starts degrading rapidly [25].

,

The state-of-the-art architectures will be trained from scratch. Then, transfer learning will be employed using the same architectures but with the weights trained on the Imagenet dataset. We anticipate the transfer learning to yield good performance since the microstructural images share common visual features including edges, blobs, visual textures with natural images. Using transfer learning will not only save time during learning but it will also be useful as we do not have ideally sufficient images to train a deep CNN.

It is important to evaluate the model performance carefully to get better generalization of its performance on the unseen data. For this, the dataset will be split into train and test set, where train set will be used to train the model and its performance will be evaluated on the test set. We will be utilizing various metrics to measure the model performance, which are described below.

Accuracy: The accuracy is generally defined as the number of correct predictions divided by the total number of predictions. This metric is useful when the dataset is well balanced, that is if the dataset is equally distributed over all the classes. However, it is not a good measure in case the data is unbalanced. For instance, if 90 % of the dataset belongs to the positive and the remaining 10% to the negative. In such a case, if the model is good at predicting the positive class but really bad for the negative class, it will be around 90 % occurrence. This means the model performance is significantly biased. Therefore, other metrics are needed to interpret the model's overall performance.

Confusion Matrix: The confusion matrix provides more detailed information regarding the model evaluation since it exhibits both the correct predictions as well as the incorrect predictions of the model. The confusion matrix is also known as the error matrix as it visualizes

the quantity of class A samples misclassified as class B samples and vice versa. In case of binary classification, the confusion matrix can be represented as a 2x2 matrix, Figure 2.

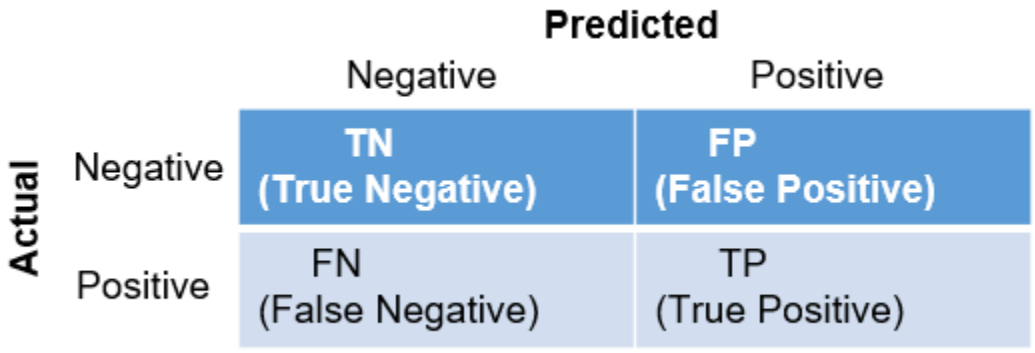

Fig 2. Confusion matrix

In the confusion matrix, each row represents an actual class, whereas each column corresponds to the predicted class. The elements along the main diagonal (that is top left to bottom right) represent correct predictions, while the off-diagonal elements are the incorrect predictions. A perfect classifier will have zero elements along the off-diagonal. Other evaluation metrics can be determined from the confusion matrix such as precision, recall, and the F1 score.

- *Precision*: This is the ratio of true positives to all the positives predicted by the model. Mathematically, it is expressed as;

$$Precision = TP/(TP + FP)$$

This means a model has lower precision if it predicts more false positives which classifies negative classes as positive classes.

- *Recall*: The recall is the ratio of true positives to the total number of actual positives. This is also called true positive rate (TPR). Mathematically,

$$Recall = TP/(TP + FN)$$

The model will have a higher recall score if it classifies more positive instances correctly. In general, increasing the precision will decrease the recall and vice versa. This is called the precision/recall trade off. Depending upon the problem at hand, one might prefer one score at the expense of another. In classifying the cancer cell as benign and malignant, for example, it is very crucial to detect all the malignant cells, hence a high recall is desired. For this task, identifying all the single phase images correctly is more important than misclassifying the multiphase images as single phase. Hence,a high value of recall is preferred.

- *F1 score*: This score provides a single score by combining both the recall and precision. A high F1 score is obtained when both recall and precision are high, that is low false positives and low false negatives. The F1 score is expressed as,

$$F1 = 2\,x\,\frac{precision\,x\,recall}{precision + recall}$$

ROC curve: The receiver operating characteristics (ROC) curve is an efficient way of evaluating the model performance by means of visualization. In the ROC curve, the true positive rate (TPR) is plotted against the false positive rate (FPR) for different decision threshold values. A dotted line with an AUC value equal 0.5 is included in the graph that represents the perfectly random classifier. The goal is then to build a classifier that stays as far away from the dotted line as possible. Thus the maximum possible value of AUC is 1 in case of a perfect classifier.

3. Dataset Preparation

The success of any ML model lies on the quality and amount of data that the model sees during the training. Data preparation is the most important step of the ML model building pipeline and thus an enormous amount of time should be spent in preparing and understanding the data before feeding it into the ML model. However, finding sufficient data to train a ML model has always challenged its performance in every field. The scarcity of data is one of the reasons for many unresolved challenges in the field of Materials Science. To perform a task similar to this project, microstructure images of both the single-phase and multiphase materials are needed. In academic research, there are more successful results published than failed attempts. Failed attempts should be reported so that ML algorithms can make useful inference from them. In fact, failed attempts are the motivation for this project; we will show how the microstructure images of multi-phase samples (this can be regarded as failed attempts) can be used to build a successful ML model. This project should motive other researchers to consider preserving the failed attempts and reporting them if possible.

As with any other material science ML projects, finding an appropriate dataset was a challenging task for this project as well. Luckily, while working in a project to discover new functional Heusler alloys over the course of the past 4 years, we have synthesized numerous alloys. Some of them turned out to be single-phase (aka successful outcome) while others exhibited multi-phase structure. After collecting data from our previous records, we found roughly 1100 images, with approximately equal examples of both classes. These images represent the sample structures that were taken using the Zeiss Axio optical microscope at different magnification levels. The samples were polished, grounded and etched by Adler reagent to reveal the microstructure. The phase-stability of all these samples were also verified by electron dispersive x-ray spectroscopy (EDS) and X-ray diffraction in addition to visualization of microstructure images. Based on the results, we are quite confident of the manual labeling of phases for the images used for this project.

Apart from images of our own research, we also collected 183 images from a few other sources including the DoITPoMS micrograph library [26]. This library houses micrographs of metals, alloys, oxides and polymers. The description is provided regarding the phase distribution, which helped with labeling these images. We also collected a few other images from the published reports, by taking screenshots of images. With the collection of images from different sources, we hope to have more variability in our dataset that better generalizes the performance of our models as compared to previous related works. In total, our dataset contains 1,283 images, which is significantly higher than the dataset used in previous works as mentioned above. We split the dataset into a train and test set, where the train contains 1107 images and the test set contains 176 images. In order to maintain variability in our test dataset, we manually generated our test set rather than using the train test split class of scikit-learn's library. Since we anticipate our model to be robust against contrast differences in the images, using the random split can not guarantee that these variations will be maintained in the dataset. This is the reason why we chose manual data split.

The images were read using openCV library and were converted into an array with three channels (i.e., with RGB format). The original image size was 1300 x 1030 pixels, but to save the memory and for faster computation we resize the image to 299 x 299 pixels (we also used 224 x 244 size to match the desired shape of pre-trained models). Both the train and test images were scaled by dividing each pixel value by 255. It should be noted here that the models were also trained using the BGR and grayscale color map, but those model's performance were significantly worse than the RGB format. Hence, we don't include those models in our discussion.

4. Model building and training

As mentioned above, we want to train several different CNN architectures and compare their performance. First, the manual model was built using 5 convolution layers, two max pooling layers, and one global average pooling layer. Each convolution layer was followed by a batch normalization layer. After the final convolution layer, a global average pooling layer was added and output of which were flattened and fed into the dense layer. In our model, we used two dense layers and a final output layer with one neuron and sigmoid activation function to make binary classification. Dropout layers were used after the dense layers with the dropout rate of 0.3.

We utilized GridSearchCV class from scikit-learn's library to find the optimizer, activation function, number of filters, number of neurons, and the learning rate. Below (table 1) we summarize the results from the hyperparameter tuning for the above mentioned architecture.

Table 1: Hyperparameter tuning for manual model

Hyperparameter	Best parameter	Accuracy
Optimizer	AdaMax	0.909
Activation function	ReLU	0.889
No. of filters (Conv.)	32, 64, 128, 256, 256	0.917
No. of neurons (Dense)	1024, 256	0.917
Learning schedule	Exponential Decay	0.938 (test data)
Initial learning rate	0.0009	0.922

Using the optimized parameters, the model was trained on the dataset for 50 epochs with a batch size of 16. During training 0.15 % of the dataset was separated for the model validation.

In addition to the aforementioned model, we trained several state-of-the-art architectures such as AlexNet, GoogleNet, VGG16, and ResNet. Since our dataset is fairly small, training these deep architectures from scratch did not produce better results. Therefore, we switched to transfer learning which involved using the weights trained on Imagenet dataset. For transfer learning, we trained three models viz.: Inceptionv3, VGG16, and ResNet50. The top layers of these pretrained were removed. The global average pooling layer was then added to the output followed by a few dense layers and a final dense layer to make the classification. In each case, we used the same output layer architecture: Dense (1025), Dropout(0.3), Dense(256), Dropout(0.3). The ReLU activation was used for all the Dense layers except the output layer, where a sigmoid function was used with hopes to make binary classification. First, these models were trained by freezing all the pre-trained model's layers for 10 epochs. After the initial training, a few top convolution layers were made trainable so that the high level features could be modified to fit our problem. Doing so enhances the model's accuracy, which we will discuss in the next section.

Notebook containing how to access the image data, training of different models, and their evaluation is available on GitHub.

5. Model Evaluation and Conclusion

As mentioned above, the model's performance was evaluated using the test images, which were not the part of the training images. We used various metrics to access the detailed information about the model's performance. In the table 2 below, we present the metric scores that were obtained for some of the best performing models.

Table 2: Performance metrics for different models

Model	Accuracy	Precision	Recall	F1-score	AUC
Manual	0.932	0.920	0.941	0.930	0.971
Pre-trained Inceptionv3	*0.966*	*0.976*	*0.953*	*0.964*	*0.994*
Pre-trained VGG16	0.949	0.942	0.953	0.947	0.994
Pre-trained ResNet50	0.864	0.859	0.859	0.859	0.883

From the table above, it is clear that pre-trained InceptionV3 outperforms all other models. The confusion matrices and ROC curves, shown below, shed more light on the performance of different models. Note that as pre-trained ResNet performanced worse than others, we don't include its confusion matrix and the ROC curve in the figure 3 below.

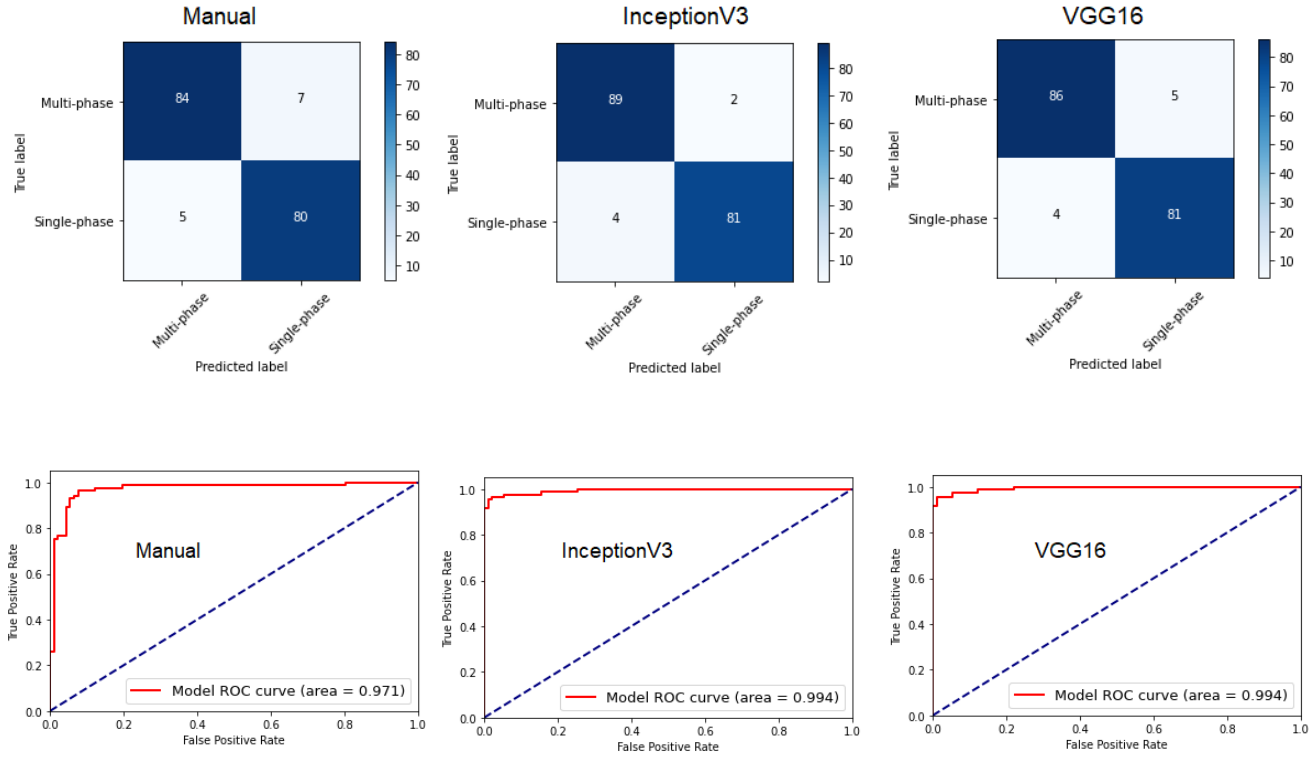

Fig 3: Top row shows confusion matrix for top three models, whereas the bottom row shows the ROC curves exhibiting the area under the curve.

All three models are relatively good in classifying the single-phase images. But, the InceptionV3 outperforms the other two when it comes to classifying the multiphase images. There are six images, which the inceptionV3 misclassified, so analyzing those images is helpful to understand the cause and potentially improve its performance. Six images are shown below (figure 4) along with the actual and predicted classes (here class 0 represents multi-phase image whereas 1 represents single-phase). The predicted probability is also included that shows probability

belonging to the positive class, i.e., 1 (we defined 0.5 as our decision threshold to make class prediction). Image (a) was misclassified as a multi-phase image, but its probability is 0.45, i.e., close to the threshold. That means the model is 45% confident that the image is single-phase, which is a decent prediction. For other images, the prediction is pretty bad. For example, the model is 97% confident that image (b) is single-phase, which is incorrect. However, from the analysis of these images, we can conclude that our model is still not strong enough to learn the complex contrast pattern. One reason for this could be we did not have sufficient images to train the models properly. Further work involves implementing more sufficient images involving complex contrast patterns. Nonetheless, so far our results suggest that our models have higher accuracy than the previous related works, which can be regarded as a remarkable accomplishment.

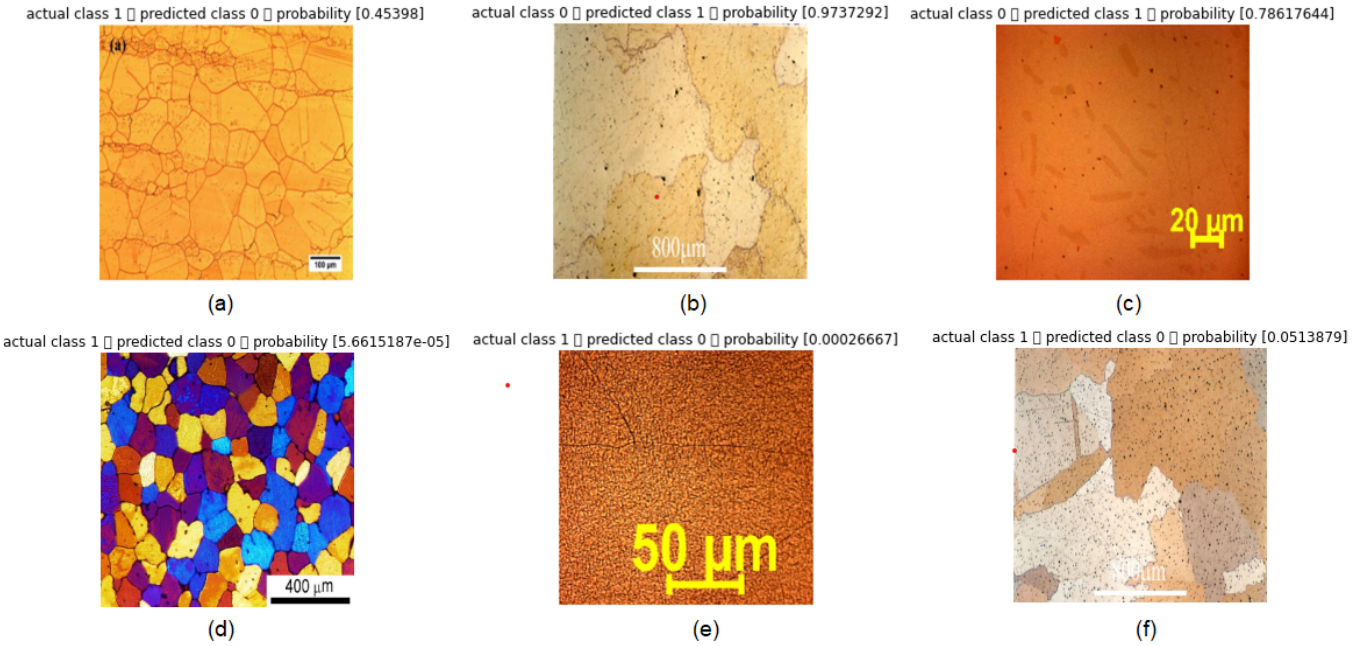

Fig 4: Misclassified test images by the InceptionV3 model.

6. Model Deployment

After having a successful model, it is satisfying to have a publicly available platform for anyone interested to upload their image for classification of the phase-stability of their material. Motivated by this, we decided to make a web application and deploy it. The general idea is the user will be able to upload an image and check if they have single-phase or multiphase microstructure, based on the prediction of the deep learning model. Up until writing this report, we have successfully built an app that takes an image and makes predictions about the phase distribution. This app was built using Django framework, a python based free and open source web framework. Snippets showing how this app works is shown in figure 5. The left image shows the user interface, where they can upload an image from their local machine and submit it to make predictions. The image on the right shows the uploaded image and the predicted outcome (i.e., whether the image is single-phase or multiple phase). The user can upload as

many images as they like by clicking the classify another image button. Unfortunately, due to memory limitations, we are unable to make this application live (the website containing this app runs in heroku server, which allows a maximum slug size of 500 MB). The snapshots shown in figure 5 are from the app running in the local host. However, as we are working on fixing the memory issue, if everything works out this app will be available at www.shambhukc.com/resources. Alternatively, we are also thinking about pursuing other platforms like streamlit, which offers more user friendly features and is optimized for deploying machine learning models.

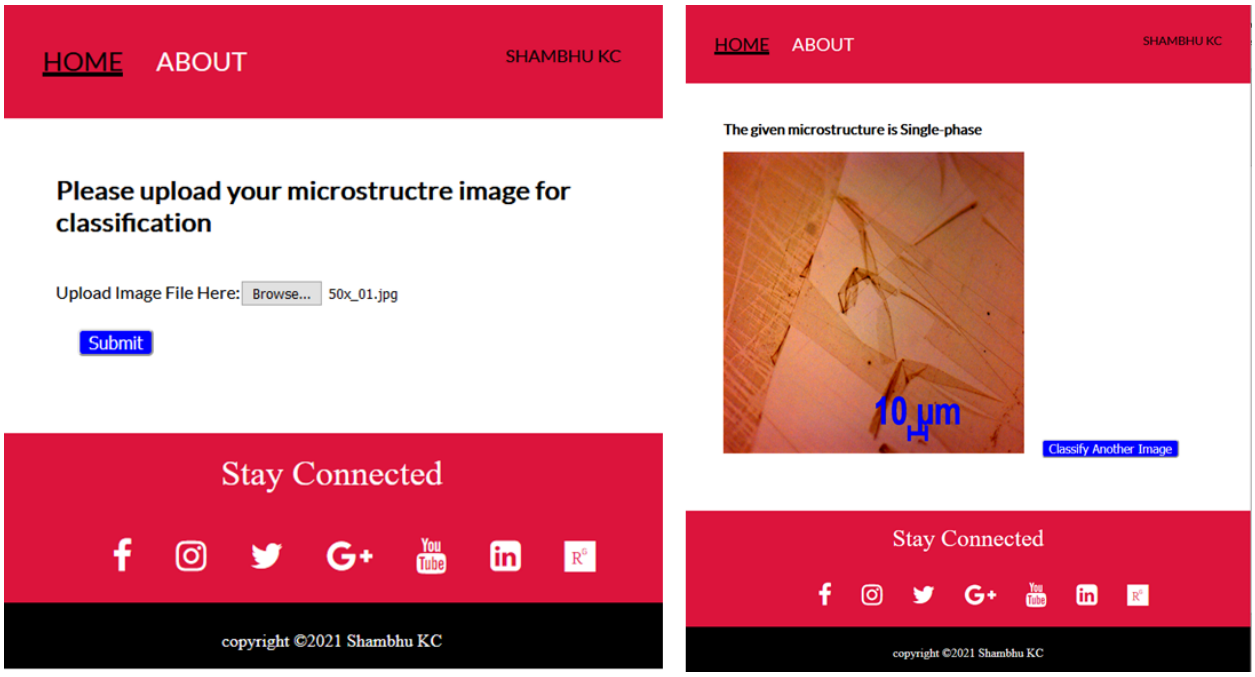

Fig 5: Snapshots of web application which uses InceptionV3 model to make phase prediction.

7. Future Direction

In this project, we successfully trained and validated different CNN architectures to classify the microstructure of alloys. A high accuracy is achieved for all the models, although the pretrained models seemed to produce better results. We noticed that the models are still not robust enough to learn complex contrast patterns. One possible solution for this issue is to increase our train dataset. This is our main priority as we move with this project.Therefore, collaboration with material engineering research teams at UA are in process to accomplish this goal. As those groups work with different classes of alloys, not just Heusler alloys, we expect to have more variety in our dataset in addition to expanding the dataset size.

After the classification of images, one may, however, be interested in knowing the distribution of phases in case of multi-phase microstructure. Moreover, he/she might want to estimate the phase fraction, i.e., area covered by the different phases. There are a couple of commercially available software such as ImageJ, which can be used for such tasks. However, a

significant manual effort is required in such a case, as it requires tedious and time consuming pixel wise labeling of different phases for each image. It would be beneficial if a model could perform instance segmentation as well as estimate the area covered by the instances without requiring manual effort (i.e., instance segmentation with regression) was available. This idea sounds novel and astounding, obtaining such a goal is not that easy. First, it is necessary to prepare the training dataset with instance masking which is a lengthy process. Then, which architecture to use needs to be determined. Fortunately, there are a couple of algorithms that are already available such as Mask R-CNN, slightly modifying it to perform regression tasks can be beneficial. Hence, assuming that the training data is available, achieving such a task sounds feasible. Therefore, this is an interesting task for our future projects.

References

1. RA De Groot, FM Mueller, PG Van Engen and KHJ Buschow. “New class of materials: half-metallic ferromagnets”. In:Physical Review Letters 50.25 (1983),p. 2024.doi:10.1103/PhysRevLett.50.2024.
2. JH Wernick, GW Hull, TH Geballe, JE Bernardini and JV Waszczak. “Super-conductivity in ternary Heusler intermetallic compounds”. In:Materials Letters 2.2 (1983), pp. 90–92.doi:10.1016/0167-577X(83)90043-5.
3. Stanislav Chadov, Xiaoliang Qi, J ̈urgen K ̈ubler, Gerhard H Fecher, Claudia Felser and Shou Cheng Zhang. “Tunable multifunctional topological insulators in ternary Heusler compounds”. In:Nature materials 9.7 (2010), pp. 541–545.doi:10.1038/nmat2770.
4. Olga Meshcheriakova, Stanislav Chadov, AK Nayak, UK R ̈oßler, J ̈urgen K ̈ubler,G Andr ́e, AA Tsirlin, J Kiss, Steffen Hausdorf, Adel Kalache et al. “Large non-collinearity and spin reorientation in the novel Mn_2RhSn Heusler magnet”. In:Physical review letters 113.8 (2014), p. 087203.doi:10.1103/PhysRevLett.113.087203.
5. C Uher, J Yang, S Hu, DT Morelli and GP Meisner. “Transport properties of pure and doped MNiSn (M= Zr, Hf)”. In:Physical Review B 59.13 (1999),p. 8615.doi:10.1103/PhysRevB.59.8615.
6. Scott A Chambers and Young K Yoo. “New materials for spintronics”. In:MRSbulletin 28.10 (2003), pp. 706–710.doi:10.1557/mrs2003.210.
7. David Kieven, Reiner Klenk, Shahab Naghavi, Claudia Felser and Thomas Gruhn.“I-II-V half-Heusler compounds for optoelectronics: Ab initio calculations". In:Physical Review B 81.7 (2010), p. 075208.doi:10.1103/PhysRevB.81.075208.
8. Tanja Graf, Claudia Felser and Stuart SP Parkin. “Simple rules for the understanding of Heusler compounds”. In:Progress in solid state chemistry 39.1(2011), pp. 1–50.doi:10.1016/j.progsolidstchem.2011.02.001.
9. Uichiro Mizutani. “The Hume-Rothery rules for structurally complex alloy phases”. In:Surface Properties and Engineering of Complex Intermetallics. World Scientific, 2010, pp. 323–399.doi:10.1142/9789814304771_0011.
10. David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, GeorgeVan Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneer-shelvam, Marc

Lanctot et al. “Mastering the game of Go with deep neural net-works and tree search”. In:nature 529.7587 (2016), pp. 484–489.doi:10.1038/nature16961.

11. Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. “Delving deep into Rectifiers: Surpassing human-level performance on imagenet classification”. In:Proceedings of the IEEE international conference on computer vision. 2015,pp. 1026–1034.doi:10.1109/ICCV.2015.123.
12. MG Schultz, C Betancourt, B Gong, F Kleinert, M Langguth, LH Leufen, Amir-pasha Mozaffari and S Stadtler. “Can deep learning beat numerical weather prediction?” In:Philosophical Transactions of the Royal Society A379.2194 (2021),p. 20200097.doi:10.1098/rsta.2020.0097.
13. Turab Lookman, Stephan Eidenbenz, Frank Alexander and Cris Barnes .Materials discovery and design: By means of data science and optimal learning.Vol. 280. Springer, 2018.doi:10.1007/978-3-319-99465-9.
14. Kyoungdoc Kim, Logan Ward, Jiangang He, Amar Krishna, Ankit Agrawal andC Wolverton. “Machine-learning-accelerated high-throughput materials screen-ing: Discovery of novel quaternary Heusler compounds”. In:Physical ReviewMaterials 2.12 (2018), p. 123801.doi:10.1103/PhysRevMaterials.2.123801.
15. Wei Li, Ryan Jacobs and Dane Morgan. “Predicting the thermodynamic stability of perovskite oxides using machine learning models. In:Computational Materi-als Science 150 (2018), pp. 454–463.doi:10.1016/j.commatsci.2018.04.033.
16. James Nelson and Stefano Sanvito. “Predicting the Curie temperature of fer-romagnets using machine learning”. In:Physical Review Materials 3.10 (2019),p. 104405.doi:10.1103/PhysRevMaterials.3.104405.
17. Yun Zhang and Xiaojie Xu. “Machine learning modeling of lattice constants for half-Heusler alloys”. In: AIP Advances 10.4 (2020), p. 045121.doi:10.1063/5.0002448.
18. Han Wei, Shuaishuai Zhao, Qingyuan Rong and Hua Bao. “Predicting the effective thermal conductivities of composite materials and porous media by machine learning methods”. In:International Journal of Heat and Mass Transfer 127(2018), pp. 908–916.doi:10.1016/j.ijheatmasstransfer.2018.08.082.
19. Randy Jalem, Kenta Kanamori, Ichiro Takeuchi, Masanobu Nakayama, Hisat-sugu Yamasaki and Toshiya Saito. “Bayesian-driven first-principles calculations for accelerating exploration of fast ion conductors for rechargeable battery application”. In:Scientific reports 8.1 (2018), pp. 1–10.doi:10.1038/s41598-018-23852-y.
20. Brian L DeCost and Elizabeth A Holm. “A computer vision approach for auto-mated analysis and classification of microstructural image data”. In:Computa-tional Materials Science. 110 (2015), pp. 126–133.doi:10.1016/j.commatsci.2015.08.011.
21. Aritra Chowdhury, Elizabeth Kautz, B ̈ulent Yener and Daniel Lewis. “Image driven machine learning methods for microstructure recognition”. In:Computa-tional Materials Science 123 (2016), pp. 176–187.doi:10.1016/j.commatsci.2016.05.034.
22. Dali Chen, Dinghao Guo, Shixin Liu and Fang Liu. “Microstructure instance seg-mentation from aluminum alloy metallographic images using different loss func-tions”. In:Symmetry 12.4 (2020), p. 639.doi:10.3390/sym12040639.
23. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, DragomirAnguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich. “Going

deeper with convolutions". In:Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, pp. 1–9.
24. Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In:arXiv preprint arXiv:1409.1556(2014).
25. Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. "Deep residual learn-ing for image recognition". In:Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 770–778.
26. https://www.doitpoms.ac.uk/miclib/index.php