Identify Material Degradation by Deep Learning Approach

STAT 6650
Team Member: Yunli Zhang, Rui Shi

Abstract

The degradation of polymer is an issue for reliability engineering. The material degradation could be characterized by mechanical properties and microstructure evolution. However, though the microstructure evolution and mechanical properties are related, their relationship is hard to quantify by physical approach. Thus, we are trying to reveal the connection between mechanical properties and microstructure evolution in a deep learning model. In this study, the deep learning method is applied to investigate the crack generation of the material. Image classification techniques are applied.

1. Introduction

Underfills encapsulants are widely used in microelectronic industry to improve the reliability of FCBGA from the thermal shock and mechanical shock (错误!未找到引用源。). UF would be placed in the gap of silicon ship and PCB, help solder balls support the chip. What is more, UF could also decrease the CTE mismatch. As the result, the stability of UF material would significantly affect the reliability of microelectronic package. As the electronic system becomes more and more powerful, the electronic package would have to stay in the high temperature environment for long period of time. Currently, most electronic components are manufactured for working under 55-100℃ for around 3-5 years. The information about UF working in higher temperature and longer time is not available. Thus, it is significant to investigate the degradation mechanisms for UF material stored in the high temperature for a long time. In this project, we investigate using convolutional neural network.

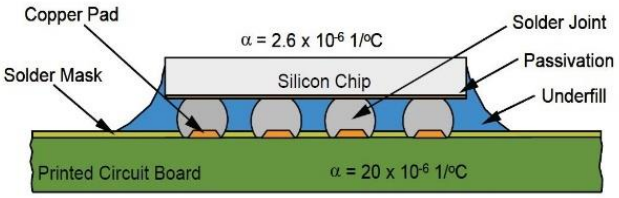

Figure 1 Typical Assembly BGA

First of all, we prepared a number of UFs materials. Uniaxial tensile tests are conducted to study the mechanical properties evolution as the function of aging time. After that, optical microscope was used to characterize the microstructure evolution as the function of aging time. The images of cross-section area of uniaxial tensile samples[1]. The evolution of material boundary layer is shown below Figure 2.

In order to capture the samples with and without crack, we select the images of samples with cracks from the pool of images. The images with cracks are shown in Figure 3.

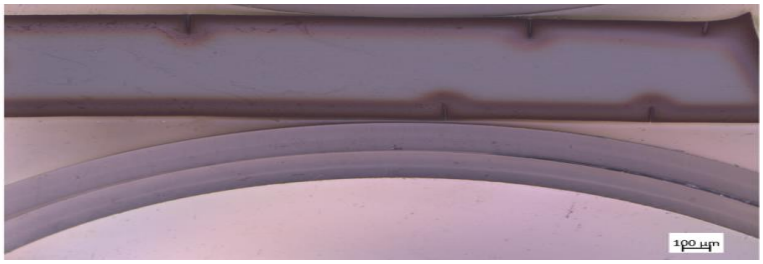

Figure 2 Boundary Layer for UFs Material

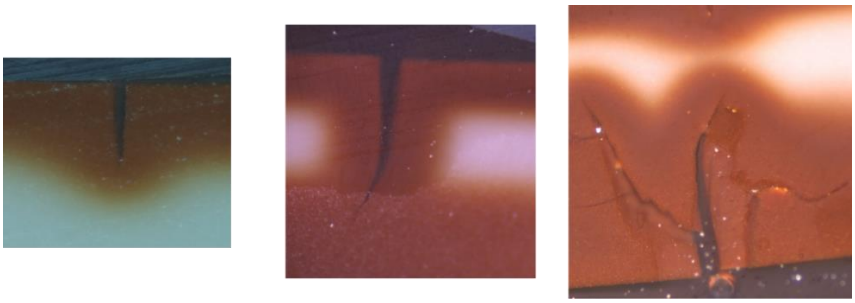

Figure 3 Samples with Cracks

The images without cracks are put in another folder. The images without crack are shown Figure 4:

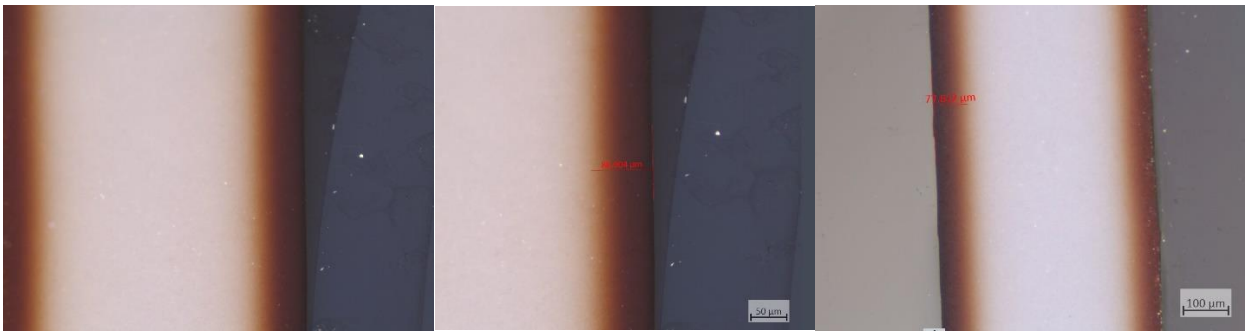

Figure 4 Samples without Cracks

At last, we have 58 images with cracks and 192 images without cracks. The small samples image classification techniques are conducted to learn the cracks. It is supervised learning.

2. Method and analysis

2.1 Graphics: Digital Image Processing

Digital image processing is the use of computer algorithms to manipulate digital images. This technique has a wide range of uses, from digital photography to applications in data compression, computer graphics, computer vision, and robotics.

A digital image is made of "pixels" arranged in a grid of rows and columns. You don't typically see the individual pixels, but they are there (see Figure 5). An image on a computer screen appears smooth or continuous because very small points on the screen are used to reproduce the individual pixels.

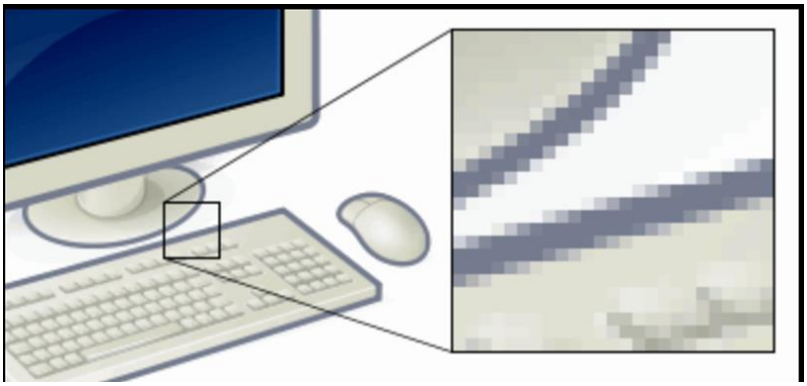

Figure 5 A image and its pixels

The pixels store data representing a color from the visible spectrum. There are different ways to specify a color, but the most common is to use the discrete RGB color model, where the individual colors are specified by the amount of red, green, and blue light needed to produce the given color. The values are given as integers between 0 (no light present) and 255 (maximum amount present). Filtering an image results in the color component values of each pixel being modified in some way. For example, you can darken a bright image, create the negative of an image, or convert an image to grayscale. Figure 6 approximates the results of applying several common filters to a variety of RGB values.

In this way, each image can be represented by three image matrices with dimension equals to it's length and width of pixels. The three image matrices correspond to the level of red, green and blue respectively.

RGB Values	15% Darker	Negative	Grayscale
255, 255, 255	217, 217, 217	0, 0, 0	254, 254, 254
0, 0, 255	0, 0, 217	255, 255, 0	18, 18, 18
128, 128, 128	109, 109, 109	127, 127, 127	128, 128, 128
0, 255, 0	0, 217, 0	255, 0, 255	182, 182, 182
255, 0, 0	217, 0, 0	0, 255, 255	54, 54, 54
35, 178, 200	30, 151, 170	220, 77, 55	149, 149, 149
255, 255, 0	217, 217, 0	0, 0, 255	236, 236, 236
0, 255, 255	0, 217, 217	255, 0, 0	200, 200, 200

Figure 6 Sample RGB Values

2.2 Brief Intro on Artificial Neural Network (ANN)

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

Before we talk about artificial neural network, we need to talk about the single neuron first. The structure of single neuron is showed in the following figure 6. A single neuron has multiple weight

adjusted inputs, those inputs will be processed by an activation function with threshold. The output of it will be connected to the next neural with a weight.

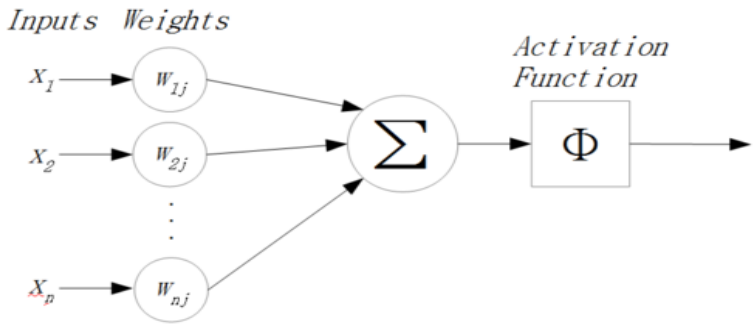

Figure 7 The structure of a single neuron

Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. This can be showed in figure 8.

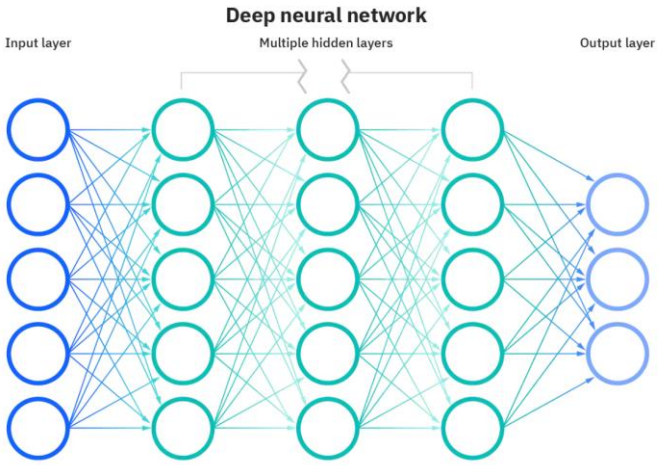

Figure 8 The structure of a single neuron

Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity. Tasks in speech recognition or image recognition can take minutes versus hours when compared to the manual identification by human experts.

However, when doing image recognition, ANN can't take the images directly as the input, it needs to flat the three image matrices, figure 9 is an example. Thus, the pixel spatial dependencies can't be learned by ANN.

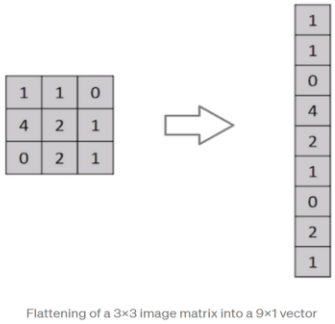

Figure 9 Flattening of a 3x3 image matrix into a 9x1 vector

Nevertheless, we can try ANN to identify the number of cracks in our material images. The number of cracks varies from 0 to 6. Here we use an ANN of four layers with 128, 64, 32, 7 neurons respectively. The training accuracy is about 0.64 after 20 iterations, which is not good enough.

```
##Traditional neural networks
model = Sequential([
  layers.experimental.preprocessing.Rescaling(1./255, input_shape=(300, 300, 3)),
  layers.Flatten(),
  layers.Dense(128, activation='relu'),
  layers.Dense(64, activation='relu'),
  layers.Dense(32, activation='relu'),
  layers.Dense(7)
])
```

Figure 10 The ANN model we used for classification.

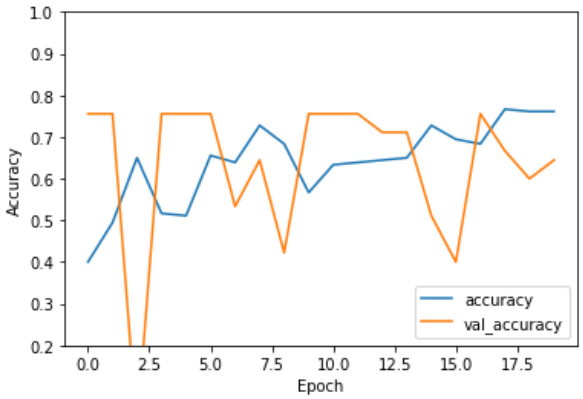

Figure 11 The experimental result for ANN

Thus, we need a more advanced model over ANN, which is the convolution neural network (CNN). We just need to add a few so-called convolutional layers over ANN. By doing this, the training error accuracy will increase dramatically to about 0.89.

```
## CNN model
model = Sequential([
  layers.experimental.preprocessing.Rescaling(1./255, input_shape=(300, 300, 3)),
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Flatten(),
  layers.Dense(128, activation='relu'),
  layers.Dense(64, activation='relu'),
  layers.Dense(32, activation='relu'),
  layers.Dense(7)
])
```

Figure 11 The CNN model we used for classification

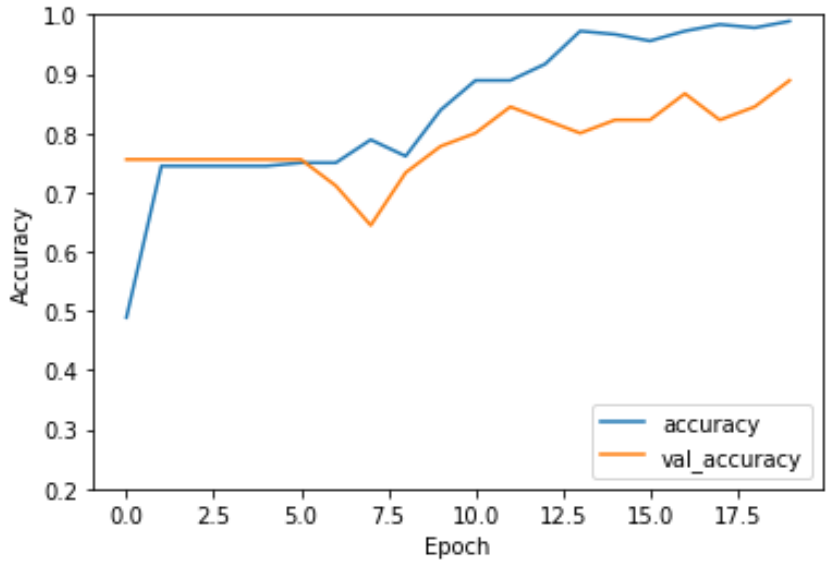

Figure 12 The experimental result for CNN

Now, we will introduce how CNN works.

2.3 The Convolutional Neural Network (CNN)

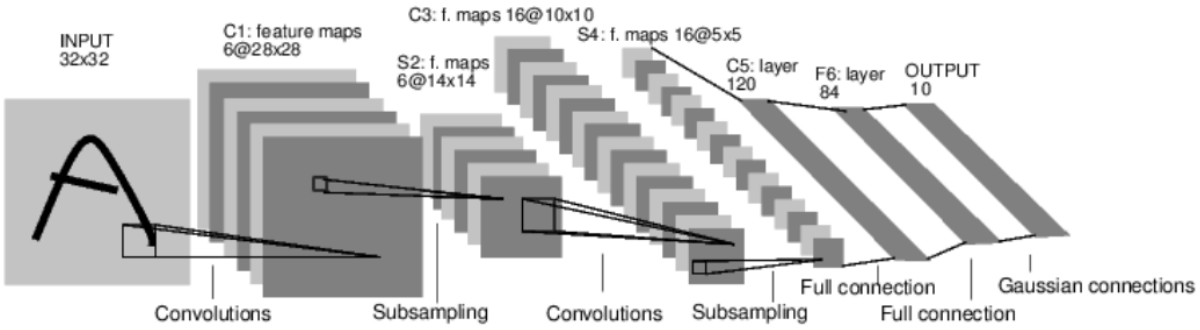

Figure 13 Structure for Convolutional Neural Network

The classic structure of convolution neural network is shown in figure 13. The convolutional neural network includes three different parts: the convolution layers, subsampling layers and fully connected layers. The convolutional layers are doing convolution operation on images. The subsampling layers filtered the information of convolutional layers and shrink it into smaller size. The convolutional layers and subsampling layers are always paired. After multiple pairs of convolutional layers and subsampling layers, there are fully connected layers, which is as the same as classical neural network.

In order to apply the convolutional neural network, we set up the TensorFlow and the environment firstly, which is pretty complicated and challenging. We resized all images and separated the data set to 20:80. 80%, 200 images for training and 50 images for validation. We normalized all images. All pixels values are now in [0,1]. After data processing, the convolutional neural network model is built, including 3 convolutional layers with ReLU as activation function, three MaxPooling subsampling layers, one flatten layer and a fully connected layer with 128 neurons with ReLU as activation function. Categorical cross entropy is used as loss function, which is commonly used in multi-label classification problem. If it is regression problem, MSE are usually used as loss function. There are 3,989,285 parameters to train.

We plot the training and validation accuracy, training and validation loss plots.

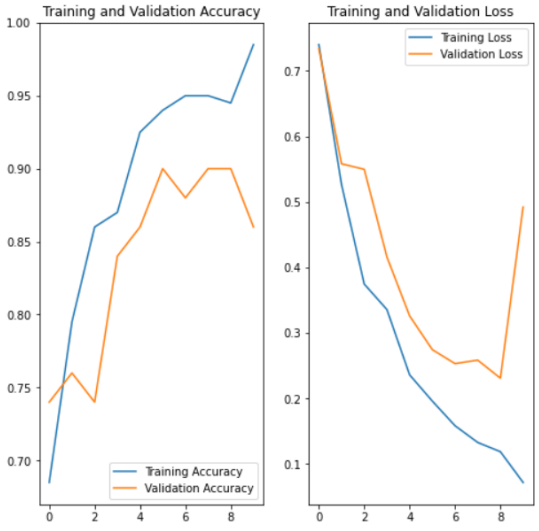

Figure 14 Experimental Results for Image Classification

For accuracy plot, the training accuracy is increasing linearly and get close to 1 after 8 epochs, which usually indicating the existence of overfitting. The validation accuracy reached about 90% then decreases. It also indicates the overfitting. For loss plot, the training loss decreases continuously as the epochs increases. The validation loss decreases to 0.2 then increases, indicating the overfitting.

Since we only have small samples, it is easy to have overfitting problem. In order to solve the overfitting problem. We used data augmentation and dropout techniques. Data augmentation takes the approach of generating additional training data from your existing examples by augmenting them using random transformations that yield believable-looking images. This helps expose the model to more aspects of the data and generalize better. The procedure of drop out is shown in figure 15:

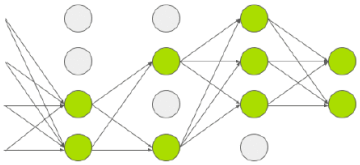

Figure 15 Drop out Technique

The dropout technique based on the philosophy that sometimes the partial is better than full. Some is better than all. It only takes part of the model to train. After applied these two techniques, the accuracy and loss graphs are showing in figure 16.

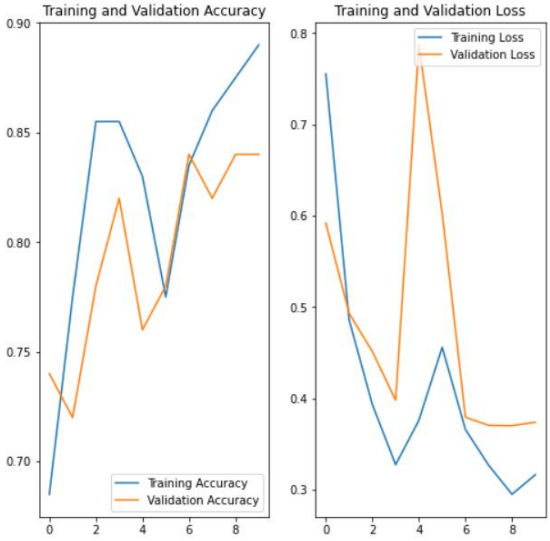

Figure 16 Results with Overfitting Improvement

The training accuracy shows the nonlinear behavior, which indicates the improvement of overfitting problem. The validation accuracy reaches about 85 percent at last. For loss plot, the trend of training and validation loss are fitted well, indicating the improvement of overfitting problem.

3. Conclusion

In this project, 250 images have been input as small samples classifications. The samples with and without cracks are recognized. ANN and CNN could be successfully implemented as classifier for small sample material classification problems. The sample features are significant for classification accuracy.

4. References

[1] The Degradation Mechanisms of Underfills Subjected to Isothermal Long-Term Aging
[2] Python for Everyone, 2nd edition, Cay Horstmann Rance Necaise