Accelerated Crystal Plasticity Simulations for Deformation Behaviour using Machine Learning

Kshitij Bagga*, Shirish Chandrakar, Swati Mahato, N. P. Gurao
Department of Materials Science and Engineering, Indian Institute of Technology Kanpur, Kanpur-208016

ABSTRACT

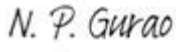
Crystal plasticity refers to the study of plastic deformation in single-crystal and polycrystalline materials while attempting to take into account explicitly the details of the physics and geometry of deformation at the crystal level. Various studies have been carried out to study the crystal plasticity of various kinds of microstructures. Several experimental techniques like In-situ EBSD and computational approaches such as CP-FFT (crystal plasticity fast Fourier transform) and CPFEM (crystal plasticity finite element method) are used in these studies. These approaches are time consuming, require advanced characterization tools, high computational costs, and demand multiple domain expertise including that of material mechanics to crystallography. To this end, the current study aims to accelerate crystal plasticity simulations using machine learning (ML). A combinatorial approach using crystal plasticity simulations using machine learning is, thus employed to predict the plasticity behaviour at higher strain. For a series of incremental tensile strains applied to simulated microstructure, material response in terms of its Euler angles and kernel average misorientation (KAM) distributions is collected. The collected datasets are further used to train the various ML Models to estimate these response parameters for any strain step beyond the 0.2 strain step. The predicted results will be further analyzed to elucidate the important parameters related to the underlying deformation mechanism. The work will be extended to different strain paths to further understand the mechanical response at different loading conditions

KEYWORDS

Crystal Plasticity, Electron Back Scatter Diffraction, Machine Learning, CPFFT, CPFEM

*Corresponding author e-mail: kshitijb22@iitk.ac.in

Mentor name and signature

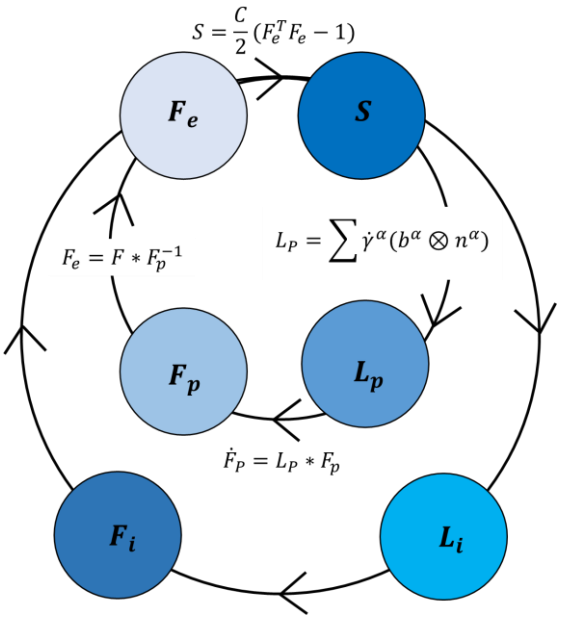


Dr. N. P. Gurao

INTRODUCTION

Crystal plasticity refers to the study of plastic deformation in crystalline materials, taking into account the physics and geometry of deformation at the crystal level. It involves understanding the different modes of plastic deformation, such as slip and twinning, and their influence on the material's stress-strain response and microstructure evolution. Since the beginning of crystal plasticity studies, real crystals or microstructures were used. It was only in the late 1990s or early 2000s, that the idea of generating synthetic microstructures was introduced and came at the stage of implementation. Dr. Anthony Rollett, a prominent figure in materials science, played a significant role in developing and promoting the use of synthetic microstructures. These synthetic microstructures are used to study various phenomena such as grain growth, recrystallization, and mechanical properties without the need for extensive and expensive experimental procedures.

Since the development of more accurate and feasible methods of synthetic microstructure generation, various methods for simulating a wide variety of loading and strain conditions have been developed. In the present work, the software DREAM3D has been used for the generation of synthetic microstructures since it offers a highly flexible and modular workflow system. It allows the design and custom processing of pipelines to perform various tasks, such as segmentation, feature extraction, and analysis, tailored to specific research needs. Furthermore, for the simulation of different kinds of strain tests on the subject microstructure, Düsseldorf Advanced Material Simulation Kit (DAMASK) has been utilized. DAMASK is a powerful tool that enables the simulation of complex mechanical behavior in materials, providing detailed insights into the deformation processes at the microstructural level. The following diagram represents the DAMASK constitutive model:

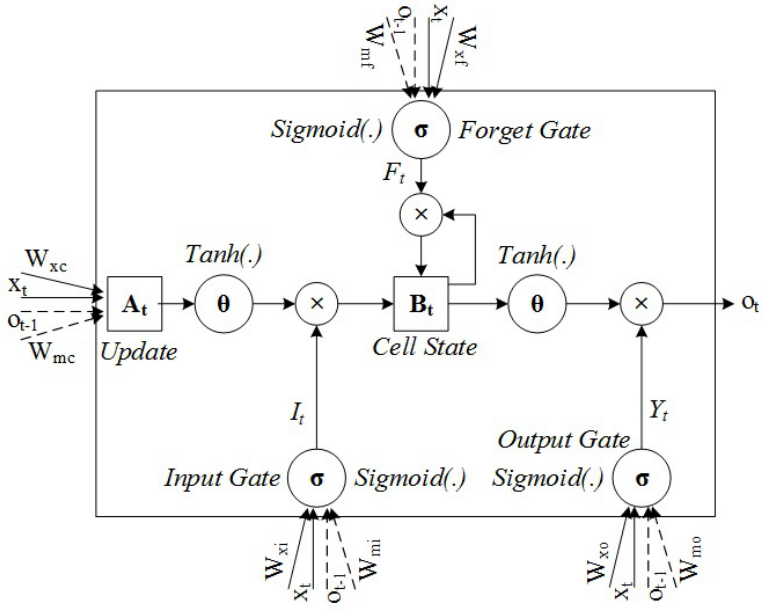


[1]

Here F is the total deformation gradient, representing the net deformation of the material. F_e is the elastic component of the deformation, i.e., the recoverable part and F_p is the non-recoverable or plastic component

of the net deformation. F_i is the intermediate configuration deformation gradient, representing the deformation between initial and final configurations. L_p and L_i are the plastic and inelastic velocity gradients respectively. S is the stress tensor, representing internal forces per unit area within the material. The equation $F_e = F.F_p^{-1}$ indicates that the elastic part of the deformation gradient can be derived by removing the plastic part from the total deformation. These concepts are fundamental in crystal plasticity modeling, where the deformation of a crystalline material is decomposed into elastic and plastic parts to better understand and predict its mechanical behaviour.

Deformation test simulations successfully provide the material parameters under different strain conditions. But in many cases, crystal plasticity involves highly non-linear and intricate relationships between microstructural features and mechanical behavior. This points to the need of other techniques to handle such complex patterns. Machine learning models can significantly reduce computation time by approximating the non-intuitive relationships involved in simulation data. Moreover, ML can leverage large datasets from experiments and simulations to make accurate predictions about material behavior without explicitly solving the underlying physical equations every time. Various ML models like Linear Regression, Polynomial Regression, Random Forest and Deep Learning (DL) models like ANN, RNN, KNN etc. are integrated with plasticity simulation techniques according to the purpose of the study. Since the current study involves a series-based input with the output parameter being dependent on a fixed number of values from the past, therefore, the Long Short Term Memory (LSTM) model is used. This model is a modified version of the classic Recurrent Neural Networks (RNN) DL model. LSTMs are designed to handle long-term dependencies effectively with their cell state and gating mechanisms. Whereas, RNNs have short-term memory and struggle with long-term dependencies due to vanishing/exploding gradients. The purpose of this study is to predict the material response parameters for extrapolated strain values (i.e. outside of the range in which the model has been trained), which suggests long-term dependencies of the output, explaining the usage of LSTM instead of RNN. The following figure encompasses the architecture of an RNN model.



[2]

In the above shown figure, x_t is the input data at time t, and o_{t-1} is the output from the previous time step *t-1*. The Forget Gate (F_t) uses a sigmoid function to decide what information from the previous cell state B_{t-1} should be discarded. The Input Gate (X_t) also uses the same function, and determines which information from the current input x_t and the previous output o_{t-1} should be added to the cell state. The Candidate Cell State (A_t) uses a *tanh function* Θ and creates a cell state that could be added to the current cell state. The Cell State (B_t) makes use of all these gates and the previous state, to define the main memory unit of the model. Further, the *Output Gate* (Y_t) determines what part of the current cell state should be the output by making use of a sigmoid function. The Final Output (o_t) makes use of a sigmoid function to combine the cell state with the output gate to return the final output.

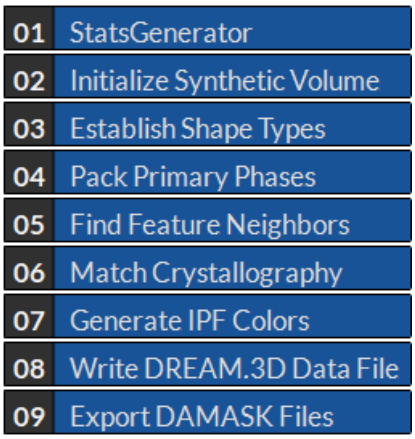
The final results that are generated using simulations and Deep Learning models need to be visualized. There are various software like *TSL-OIM* and *ATEX* that support such visualization using the generation of various maps like the Inverse Pole Figure (IPF) and the Kernel Average Misorientation (KAM) maps that are used in the project. The main advantage of using *ATEX* over *TSL-OIM* is that it is an open-source software tool designed for texture analysis. It can be downloaded and used without any licensing fees, making it accessible to a wide range of users, including academic researchers, students, and professionals in the industry. Moreover, the availability of the source code allows the users to understand the inner workings of the software and adapt it to specific research requirements.

This integrated approach combining synthetic microstructure generation, advanced simulation techniques, and machine learning predictions represents a significant advancement in the field of crystal plasticity. By accelerating the simulation process, it opens up new possibilities for the design and optimization of materials with tailored mechanical properties.

METHODOLOGY

2.1 Generation of a synthetic microstructure

The generation of a synthetic microstructure required the use of the software DREAM3D. It is a software used for creating and analyzing synthetic 3D microstructures. A 2-D microstructure of dimensions 122 x 84 was generated by constructing a pipeline that consisted of the filters in the following order.



The function of each of these filters is explained as follows:

- **StatsGenerator**: It was used to create statistical representations of the microstructure, allowing the definition grain size distributions, crystallographic textures, and phase fractions, among other parameters.

- **Initialize Synthetic Volume**: It was used to define the initial parameters for creating a synthetic microstructure volume, allowing the specification of the dimensions, resolution, and boundary conditions of the synthetic volume.

- **Establish Shape Types**: It assigned geometric shapes to the grains within the microstructure, allowing the definition of their morphology accurately. This enhances the accuracy of simulations by better mimicking the actual characteristics of material.

- **Pack Primary Phases**: It packed small or disconnected primary phases in the microstructure together, enhancing clarity by merging or removing insignificant features based on defined size thresholds.

- **Find Feature Neighbours:** It identified neighboring features within the microstructure based on the specified connectivity criteria, helping in the analysis of spatial relationships and properties between adjacent features.

- **Match Crystallography**: It is used to quantify crystallographic relationships between neighboring grains or phases within a microstructure, which is essential for understanding the orientation and interaction of crystals.

- **Generate IPF Colors**: It is used to generate Inverse Pole Figure (IPF) colors for each grain in a microstructure based on its crystallographic orientation data, enabling the analysis of grain boundaries and texture development in polycrystalline materials.

*The **Write Dream.3D Data File** & **Export DAMASK Files** filters were used to generate the output files with (.dream3d), (.config) & (.xdmf) file extensions.*

2.2 Simulating Deformation tests on the microstructure using DAMASK

The synthetically generated microstructure was then subjected to simulated incremental tensile strain using DAMASK. It employs crystal plasticity models to predict material responses under different conditions.

Using the DAMASK library in python, tensile strain was applied on the microstructure by setting the strain rate tensor as shown:

$$\begin{vmatrix} 0.001 & 0 & 0 \\ 0 & x & 0 \\ 0 & 0 & x \end{vmatrix}$$

The strain is increased incrementally till a final value of 20%. This was followed by the generation of 20 CSV files. Each of these files contains three columns- namely Phi1, Phi and Phi2 respectively. Each of them corresponds to the respective Euler Angle value for each of the 10248 (122 x 84) cells in the microstructure for that strain value (which is represented by the file index).

Similarly, these steps are performed to perform incremental compression and shear tests (both till a maximum value of 20%) on the microstructure by changing the strain rate matrix as shown below:

Compression

$$\begin{vmatrix} -0.001 & 0 & 0 \\ 0 & x & 0 \\ 0 & 0 & x \end{vmatrix}$$

Shear

$$\begin{vmatrix} 0 & 0.001 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}$$

2.3 Building a Recurrent Neural Networks (RNN) – Long Short Term Memory (LSTM) Model

The generated dataset consisted of 20 CSV files, each containing 3 columns with 10248 entries in every column. The primary purpose of the required Machine Learning model was to generate the Euler angle values for all the cells in the microstructure, for any given extrapolated strain value. Thus, an RNN-LSTM model was chosen.

A Sequential model was defined consisting of two Long Short-Term Memory (LSTM) layers followed by a Dense layer. Each of the layers are explained as follows:

- **The First Input Layer**
  model.add(LSTM(32, input_shape=(18, 1), return_sequences=True))
  This layer has 32 units and is designed to process input sequences of shape (18, 1), where 18 represents the number of time steps, and 1 represents the number of features per time step.
  The return_sequences=True parameter ensures that the output of this LSTM layer is a sequence, which is required as the input for the next LSTM layer.

- **The Second Input Layer**
  model.add(LSTM(32, return_sequences=False))
  This layer also has 32 units and processes the sequence output from the first LSTM layer.
  The return_sequences=False parameter indicates that this layer will output a single vector (the last output in the sequence), which is suitable for the subsequent Dense layer.

- **Dense Layer**
  model.add(Dense(1))
  This is a fully connected (Dense) layer with a single unit, which will produce the final output of the model. This setup is typical for regression tasks where the goal is to predict a single continuous value.

- **Compilation**
  model.compile(optimizer='adam', loss='mean_squared_error')
  There are various model compilation optimizers available- *SGD*, *momentum*, adam etc. to name a few. In the used model, “adam” is used over other optimizers since it maintains a moving average of both the gradient and its square simultaneously and thus, better minimizes the error.
  The loss function chosen is Mean Squared Error (MSE), which is appropriate for regression tasks.

2.4 Training and Testing of Data

x_train, x_test, y_train, y_test = train_test_split(X_train, Y_train, test_size=0.3, random_state=42)

The model was trained for all the three Euler angles separately. For the chosen Euler angle, a list X of dimensions 10284 x 19 is created. This list contains the 19-step strain series for each of the 10284 cells in the microstructure. A corresponding list Y of dimensions 10284 x 1 is similarly created, which contains the 20% strain value for all the cells. For the training and testing of the dataset, a train-test split ratio of 7:3 is used. The random_state parameter controls the shuffling of the data before applying the split. Setting random_state to a fixed number (e.g., 42) ensures that the split of the data into training and test sets is reproducible. This is crucial for the debugging and sharing of results

.

2.5 Prediction of the response parameters for extrapolated strain values

A *predictions* function is created which takes two arguments- *Euler* & *to_predict,* which represent the Euler angle and the strain value (in percent) for which the values of *Euler* are to be predicted. This function has been defined assuming that all the data files before *to_predict* have already been processed. A loop is run starting from 20 till *to_predict* (excluding it). The loop processes each of the 10284 series in a way such that in each iteration, the *Euler* value at the beginning of the series is removed and the value at the next strain step is appended at the end of the series. This is done so as to maintain a constant series length of 19, as defined in the model architecture. Once the loop is complete, the input for making the predictions of the values of *Euler* is ready. It finally comprises of 10284 series, each having the last 19 values in before the value to be predicted. This is followed by simply predicting the strain value at the required step and returning all those values as a list.

2.6 Generation of the Inverse Pole Figure (IPF) & Kernel Average Misorientation (KAM) maps

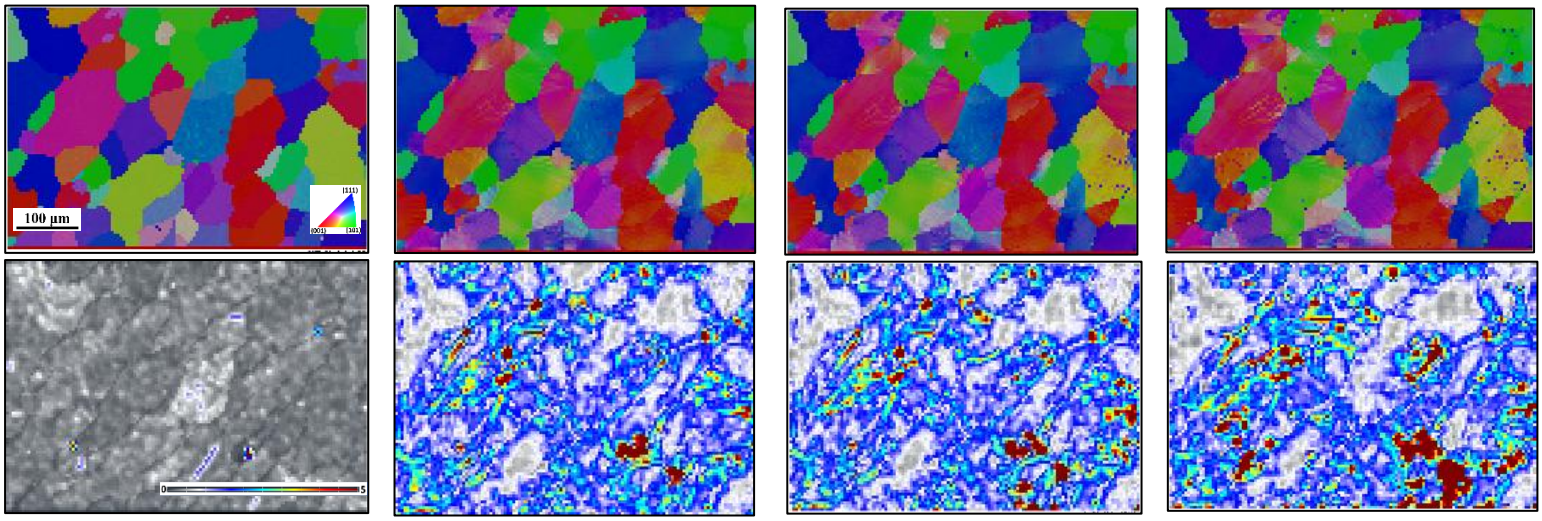
The analysis of the values generated by the LSTM model was to be done by the IPF and KAM maps. For the generation of these maps, the software ATEX was utilized. ATEX is a texture analysis software designed for characterizing and visualizing crystallographic textures in polycrystalline materials. It provides advanced tools for analyzing IPFs, KAMs and other texture-related data, enabling detailed insight into material properties and behaviors.

RESULTS

The following IPF (above) and KAM (below) maps are obtained by processing the LSTM model into ATEX:

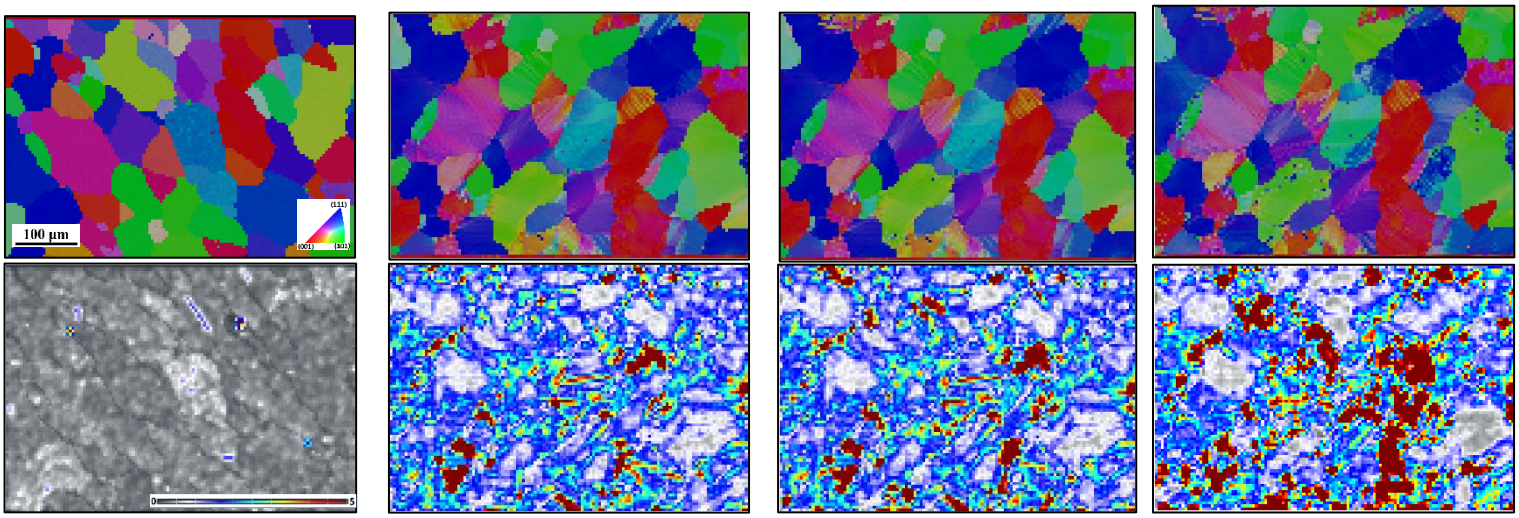
- **Tension**

Initial | 20% strained original | 20% strained predicted | 25% strained



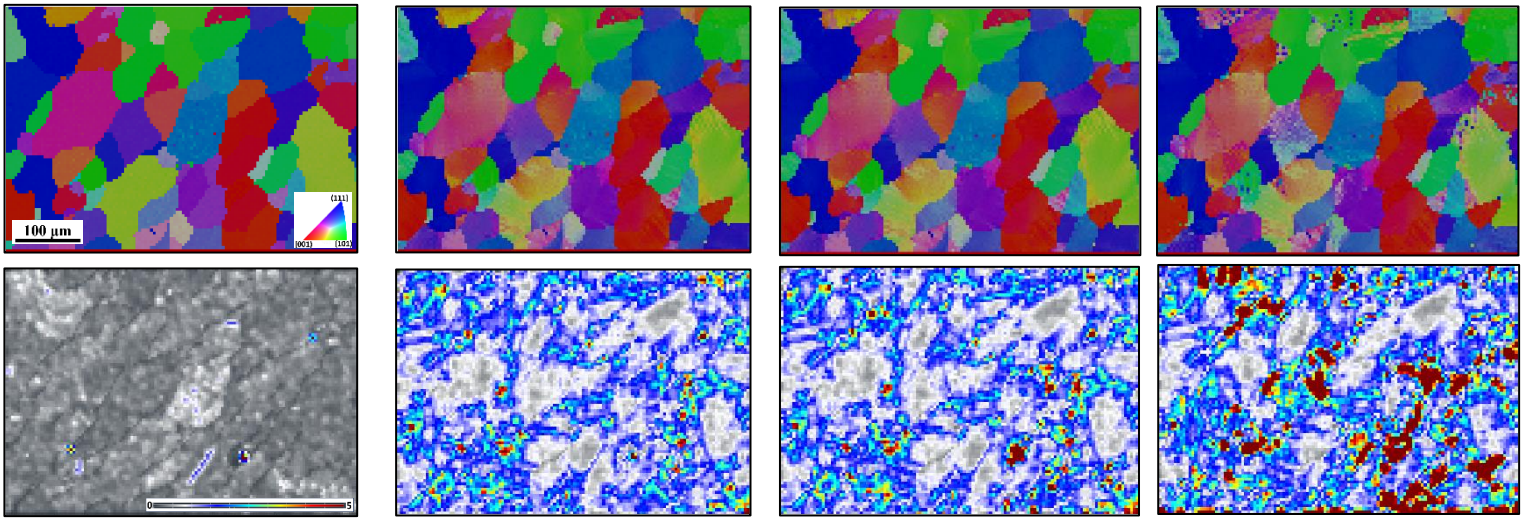
- **Compression**

Initial | 20% strained original | 20% strained predicted | 25% strained



- **Shear**

Initial | 20% strained original | 20% strained predicted | 25% strained



CONCLUSION

In the present work, a coupled simulation approach by using DREAM3D and DAMASK was used to study the texture evolution of FCC Copper microstructure when subjected ton three different strain paths- Tension, Compression & Shear respectively. Further, the RNN-LSTM Deep Learning model was used to predict the texture response of the microstructure for extrapolated strain values. From the maps shown above, it can be seen that there is a very accurate match between the simulated and predicted IPF and KAM maps for the 20% strained microstructure. The following conclusions were extracted from the obtained results:

- The texture evolution captured successfully validated with the variation mentioned in the literature.
- The evolution of high intragranular misorientation in terms of KAM is approximately captured.
- The RNN-LSTM model successfully used incremental strain values to predict the texture response at higher strain values.

FUTURE WORKS

- To build classification models that would take IPF & KAM maps as input and based on them predict the strain path that the given microstructure had been subjected to.
- To incorporate stress concentration in the given microstructure in order to predict the failure strain value.

REFERENCES

[1] F. Roters *et al.*, "DAMASK – The Düsseldorf Advanced Material Simulation Kit for modeling multi-physics crystal plasticity, thermal, and damage phenomena from the single crystal up to the component scale," *Comput Mater Sci*, vol. 158, pp. 420–478, Feb. 2019, doi: 10.1016/j.commatsci.2018.04.030.

[2] "A Novel Data-Driven Model for Real-Time Influenza Forecasting - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/An-LSTM-cell-containing-the-input-gate-the-forget-gate-and-the-output-gate-Each-gate_fig2_319619364 ".