An algorithm that can automatically detect nuclei to expedite research on a wide range of diseases, including cancer, heart disease, and rare disorders. Such tool has the potential to significantly improve the speed at which cures are developed, benefiting those suffering from various health conditions, including chronic obstructive pulmonary disease, Alzheimer's, diabetes, and even the common cold.

Hence, the identification of cell nuclei is a crucial first step in many research studies because it enables researchers to analyze the DNA contained within the nucleus, which holds the genetic information that determines the function of each cell. By identifying the nuclei of cells, researchers can examine how cells respond to different treatments and gain insights into the underlying biological processes at play. An automated AI model for identifying nuclei, offers the potential to streamline drug testing and reduce the time for new drugs to become available to the public.

Thus, you as an **AI engineer working in a biomedical company are tasked to create a model for semantic segmentation for images containing cell neuclei**.

The criteria of the project are as follows:

1. Link to the source of data: https://www.kaggle.com/competitions/data-science-bowl-2018/overview
2. Link to the dataset (please use this as your data): data-science-bowl-2018.zip
3. Carry out the whole machine learning workflow for this project (Problem Formulation →Data Preparation→Model Development→Model Deployment)
4. For criteria, try to achieve training and validation accuracy of more than 80%
5. Make sure your model is not overfitting.
6. Build a U-Net for this project (for downsampling path, you can apply transfer learning, then build your own upsampling path similarly as what we did in the exercise).
7. When uploading to GitHub, make sure your whole project is presentable:
    a. Make sure you have a complete README file.
    b. You do not need to upload the data onto GitHub, just mention the data source (which is the Kaggle link) in your README.
    c. Make sure you write your code following the good practices such as following variable naming conventions and writing concise comments to improve the readability of your code.
8. For model deployment, try to use the model to make some predictions with your test data.

Files to be submitted and uploaded to GitHub and LMS (submission link will be given on the assessment day):

1) Training, deployment scripts and classes (GitHub and LMS)
2) Saved model in .h5 format and scalers (if any) in .pkl file format. (GitHub and LMS)
3) Training process plotted using Tensorboard can be snipped and saved as image file format (LMS) and use EarlyStopping callback to prevent overfitting.
4) The architecture of the model should be plotted using plot_model function and saved as .png file format. Include the image in README.md and also upload to LMS. (GitHub and LMS)
5) Performance of the model and the reports can be snipped and saved as image file to be included in the zip folder for LMS submission. (LMS and GitHub)
6) Include your GitHub URL directing to your assessment 2 in a text file then submit to LMS. (LMS)
7) **Don't forget to credit/cite the source of the data on your GitHub page** https://www.kaggle.com/competitions/data-science-bowl-2018/overview

*Please zip all the required files into one folder then submit to LMS.

**Please save the dataset and model in 2 different folders to GitHub.

Complete the assessment and submit the files to LMS and GitHub by 5pm. Good Luck!!!

	100%	50%	0%
Task Completion (30%)	Scripts can be executed without any error on trainer’s local machine.	-	Scripts fail to be executed on trainer’s local machine.
Project requirements (30%)	Able to achieve the objectives of the project using relevant and appropriate approach.	Able to achieve the objectives of the project but using inappropriate approach such as brute forcing the solution.	Fail to achieve the objectives of the project.
Image preprocessing (30%)	Demonstrates strong understanding on the objectives of the project and performs relevant approach to process the data. Necessary image processing techniques such as, brightness correction, image filtering or image resaturation are performed and well justified.	Shows comprehensive understanding of the objectives of the project but uses incorrect or irrelevant approach to process the image.	Shows limited understanding of the objectives of the project. Absence of image processing section in the code.
Code readability (5%)	Involves the usage of functions or methods for repeated tasks. Codes are easily readable and justified by including comments and description texts.	Minimal usage of functions or methods for repeated tasks. Available comments and descriptions but lack of details.	No usage of functions or methods for repeated tasks. Codes are difficult to read and understand. Missing descriptions and comments.
GitHub repo (4%)	Detailed and clear instructions of the project on README.md. Results such as graphs are also included in README.md as part of the project description.	Project successfully uploaded to GitHub repo but with incomplete README.md. Missing descriptions, instructions, and results.	Fails to upload project to GitHub repo and missing README.md
PEP8 compliance (1%)	Fully complies with PEP 8 Standard	Partially complies with PEP 8 Standard	Fails to comply with PEP 8 Standard
Total (100%)			