******The British College**

**KATHMANDU**

**Coursework Submission Coversheet**(individual coursework only)

**Faculty of Arts, Environment and Technology LBU Student Id:**

77356702

**For checking by the student:**

Please ensure all information is complete and correct and attach this form securely to the front of your work before posting it in a coursework collection box.

Award name: BSc (Hons) Computing

Module code: COMP607

Module name: Production Project

Module run:

Coursework title: Breaking the Slience: AI and Computer Vision Driven Sign Language Translation System

Due Date:

Module leader: (In LBU)

Module Supervisor: (In TBC) Mr. Rohit Raj Pandey

**TURNITIN** Checked: YES NO ***(please circle)***

Submission date & time: Date: Time:

**Total Word Count: Total Number of Pages (including this front sheet):**

**In submitting this form with your assignment, you make the following declaration:**  
I declare, that the coursework submitted is my own work and has not (either in whole or part) been submitted towards the award of any other qualification either at LBU or elsewhere. I have fully attributed/referenced all sources of information used during the completion of my assignment, and I am aware that failure to do so constitutes an assessment offence.

Signed: ` Date:

**You are strongly advised to retain a second copy of your work in case of any query about the assignment.**

**For completion by the faculty:**

**This mark is provisional and subject to moderation and approval by the relevant examining board**

**Teacher's Feedback**

**Teacher's Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Breaking the Silence: AI and Computer Vision Driven Sign Language Translation System**

**Module: Production Project**

**Bsc. Hons. Computing**

**Level 6: 2nd semester**

1. **Introduction:**

The project, titled "Breaking the Silence: AI and Computer Vision Driven Sign Language Translation System," focuses in utilizing advanced AI and computer vision to translate sign language into both English text and speech, thereby narrowing communication gaps for deaf and mute individuals.

At its soul, the project handles the critical problem that hinder the social and professional incorporation of sign language users. By placing gesture recognition techniques applying frameworks like TensorFlow and PyTorch, alongside with hand tracking models such as Mediapipe, the system is deliberate to accurately detect and interpret hand gestures in real time. A primary aspect of the system is its user-friendly interface, which is bespoken to assure accessibility for users with limited technical knowledge. Furthermore, the blending of a text-to-speech module refines the system's functionality by furnishing audible output, further assisting efficient communication.

The system assimilates a real-time error handling mechanism to assures graceful operation by instantly warning users when a gesture is not recognized. This focus on trustworthiness and sensitivity is pivotal for practical, everyday use. Building upon previously research in sign language recognition—which grouped from hardware-dependent data gloves to modern deep learning methods—the system endeavors to subdue the exceptions posed by lively and continuing sign interpretation. It concentrates on building a scalable solution that accomplices fined under assorted conditions, such as varied lighting and hand sizes.

Eventually, the report reinforces not only the technological creations in back of the system besides also its possible social jolt. By letting clearer and more comprehensive communication, the project is composed to considerably enrich the quality of life for sign language users, enabling them to emerge more fully in educational, professional, and social surrondings..

1. **Literature Review**

Sign language Detection (SLR) refer to the process of recognizing and explanning gestures, hand movements and body language used in sign language to hold communication between people who use sign language and who do not. The field of SLR has witnessed outspoken improvements over years, staring from neural network-based systems to deep learning based systems. Early examined two stage neural network that uses a DataGlove for phoneme-level recognition, achieving 86% accuracy but was restrained by hardware depandance work (Kim, S. et. al., 1995). Another research used self-organizing frameworks like SHOSLIF-M, improving spatiotemporal recognition with a 96% accuracy rate. However, its reliance on custom-built features and limited scalability to real-life scenarios laid challenges (Cui, Y. et. al., 1995). Another Research focus on static, isolated, single-handed signs using camera-based systems, featuring a lack of organized datasets and a need for dynamic sign recognition improvements (Wadhawan, A. et. al., 2021). The use of CNNs showed odd performance with 99.90% accuracy for static signs using extensive datasets, highlighting the strength of deep learning (Kumar, P. et. al., 2020). Use of statistical methods for nonstop sign language handle real-life variability, placing multimodal features like facial landmarks and achieving revealing word error rate reductions (Koller, O. et. al., 2015). Another Study used statistical methods and fenones for efficient sign recognition but strumbled with sturdy subunit definition for varied gestures (Bauer, B. et. al., 2002). Another studyimproved a real-time sign language interpreter using a data glove and HMMs but met challenges with endored dependency and limited accuracy (Ouhyoung, M. et. al., 1998).

1. **Review Of Technology:**

The project adopts a methodical and research-driven approach, beginning with an extensive review of existing sign language recognition (SLR) technologies. This initial research helps identify limitations in past systems and informs the selection of suitable models and frameworks for the project. A crucial early step involves choosing a dataset, such as the American Sign Language (ASL) dataset or creating a custom dataset with both static and dynamic signs. These datasets form the basis for training machine learning models capable of recognizing a wide range of gestures. To support gesture detection, the project uses reliable pre-trained models like Mediapipe for real-time hand tracking, while TensorFlow and PyTorch are employed to build and train classification models using Convolutional Neural Networks (CNNs) for static gestures and potentially Recurrent Neural Networks (RNNs), LSTMs, or Transformers for dynamic gesture recognition.

Each system component is developed and tested independently to ensure performance, responsiveness, and reliability before full integration. Key metrics such as accuracy, latency, and robustness in various environmental conditions—like changes in lighting and hand orientation—are monitored throughout the testing process. This modular and iterative development strategy enables ongoing improvement at each stage and ensures the final system can perform consistently in real-world scenarios. The methodology prioritizes adaptability and modularity, allowing for early identification and resolution of issues while refining model performance based on observed results and user feedback.

To enhance accessibility, the project integrates a Text-to-Speech (TTS) module using the pyttsx3 library, enabling real-time conversion of recognized gestures into spoken English. A simple, intuitive user interface is also planned, ensuring the system can be used by individuals with limited technical expertise. The system is designed to provide meaningful feedback during use, further improving the user experience. This comprehensive and iterative methodology, grounded in continuous testing, user-centered design, and flexible development, ensures that the final product will be accurate, inclusive, and effective in helping deaf and mute individuals communicate more easily with others.

1. **Methodology:**

To manage the complexity and evolving requirements of the project, an Agile methodology is adopted to guide the development process. Agile is particularly well-suited for applications involving artificial intelligence and computer vision, where continuous testing, model refinement, and user feedback play a crucial role. By breaking the project into smaller, manageable iterations or sprints, the team can focus on building and evaluating individual components—such as hand tracking, gesture recognition, and text-to-speech translation—before integrating them into the larger system. This allows for early identification of issues, ongoing improvements, and flexibility to adjust the project based on testing outcomes or technical challenges.

Each sprint in the Agile workflow includes stages of planning, development, testing, and review. For example, a dedicated sprint may focus on optimizing gesture classification using TensorFlow or PyTorch, where performance is assessed through metrics like recognition accuracy and processing speed. Once that component is validated, the next sprint may involve integrating it with the hand-tracking system powered by Mediapipe. Feedback gathered during each sprint—whether from testing or user evaluations—is used to refine both the functionality and user experience of the system. This iterative cycle ensures that the final product is not only accurate and responsive but also usable and accessible in real-world conditions.

To support the Agile approach, the project utilizes planning tools such as Gantt charts and visual timelines to define milestones, set deadlines, and monitor progress throughout the development lifecycle. These tools provide structure and clarity while allowing room for the iterative nature of Agile development. Collaboration platforms like GitHub are used for version control, while Google Drive and Microsoft Teams assist with documentation and team coordination. By combining the adaptability of Agile with effective planning and communication tools, the project ensures that development remains focused, collaborative, and responsive to change—resulting in a robust and inclusive sign language translation system.

A screenshot of a computer

Description automatically generated

Fig - 1: *The above Project Timeline shows the project planning along with the dates.*

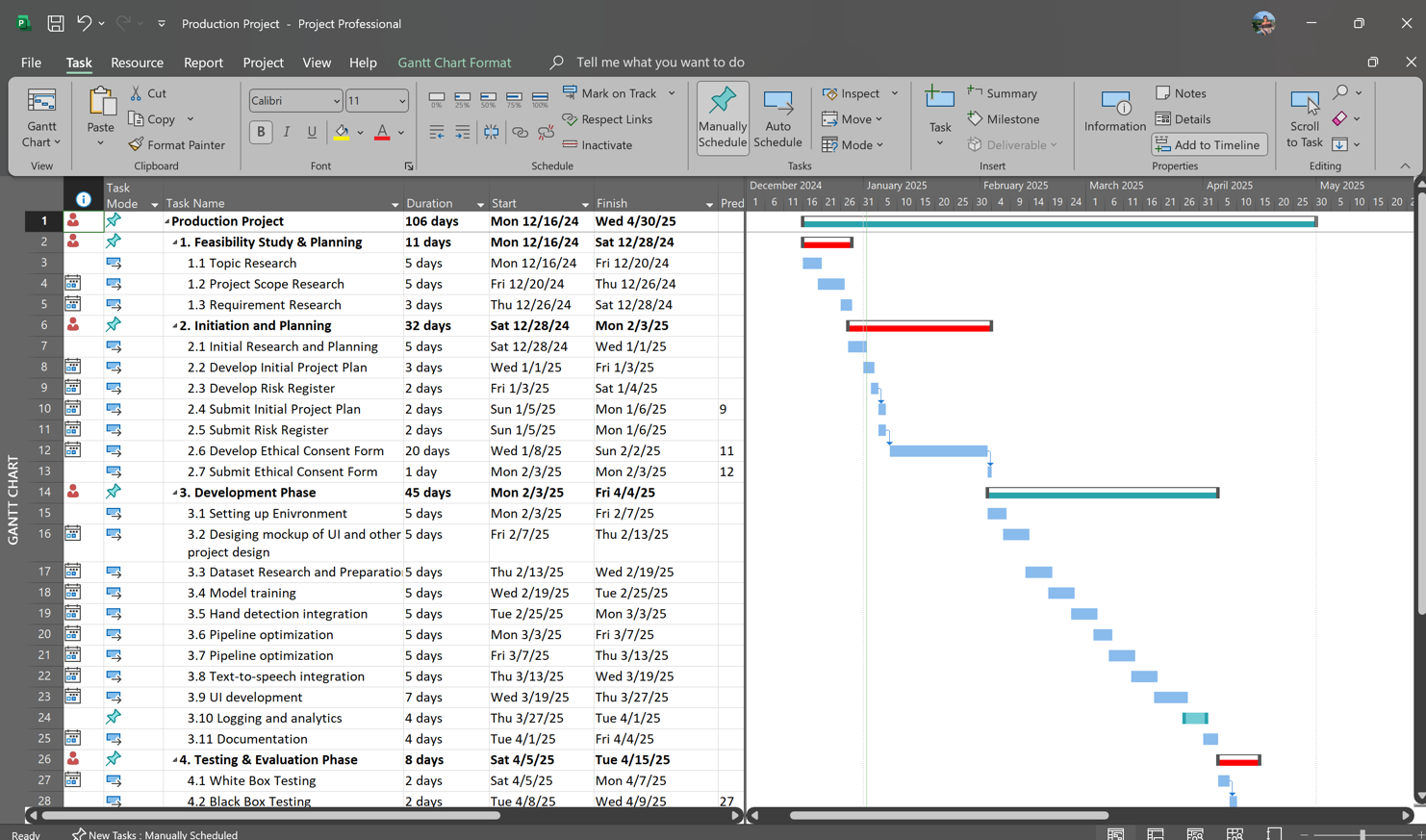


Fig - 2: *The above Gantt chart shows the project planning along with the key dates.*

1. **BIBLIOGRAPHY**
2. ***Bauer, B., Swets, D.L., Weng, J.J. Friedrich, K.K., (2002) Towards an Automatic Sign Language Recognition System Using Subunits. Springer Nature Link. [Online]. 2298, Available from: <https://link.springer.com/chapter/10.1007/3-540-47873-6\_7> [Accessed 31 Dec 2024].***
3. ***Cui, Y. , (1995) Learning-based hand sign recognition using SHOSLIF-M. IEEE. [Online]. 0 (10.1109/ICCV.1995.466879), pp. 0. Available from: <https://ieeexplore.ieee.org/abstract/document/466879> [Accessed 31 Dec 2024].***
4. ***Kim, S., Waldrom, M.B. (1995) Isolated ASL sign recognition system for deaf persons. IEEE. [Online]. 3 (3), pp. 261 - 271. Available from: <https://ieeexplore.ieee.org/abstract/document/413199> [Accessed 31 Dec 2024].***
5. ***Koller, O. (2015) Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. Computer Vision and Image Understanding. [Online]. 141, pp. 108-125. Available from: <https://www.sciencedirect.com/science/article/abs/pii/S1077314215002088> [Accessed 31 Dec 2024].***
6. ***Kumar, P. Wadhawan, A****.****(2020) Deep learning-based sign language recognition system for static signs. S.I. : Hybrid Artificial Intelligence and Machine Learning Tech. [Online]. 32 , pp. 7957–7968. Available from: <https://link.springer.com/article/10.1007/s00521-019-04691-y#citeas> [Accessed 31 Dec 2024].***
7. ***Ouhyoung, M., Ling, R.H., (1998) A real-time continuous gesture recognition system for sign language. IEEE. [Online]. pp. 10.1109/AFGR.1998.671007. Available from: <https://ieeexplore.ieee.org/abstract/document/671007/authors#citations> [Accessed 31 Dec 2024].***
8. ***Wadhawan, A., Kumar, P. (2019) Sign Language Recognition Systems: A Decade Systematic Literature Review. Springer Nature Link. [Online]. 28 , pp. 785–813. Available from: <https://link.springer.com/article/10.1007/s11831-019-09384-2> [Accessed 31 Dec 2024].***