# Project Proposal: Deep learning acoustic keystroke transcriber

**Choice of dataset:**

We plan to use Kaggle's database on keystroke sounds.

However, bearing in mind the potentially low reproducibility of the experiment (the dependence on a specific keyboard, microphone, typing patterns, language) it is important for us to train our model on a keyboard that is readily available to us. Therefore, we may also consider creating our own database. In [1], the authors designed a side-channel attack by using each of the 36 of a laptop's keys (0-9, a-z) with each being pressed 25 times in a row, varying in pressure and finger, and a single file containing all 25 presses from two standpoints: through a zoom call operated on the laptop and through a mobile phone placed nearby (60cm away). This could be easily reproduced for our final project.

**Methodology:**

The Kaggle dataset is a collection of many audio clips containing a single key press on a keyboard for each letter and number. This could constrain us when testing because we don't have access to the same keyboard used to produce the sounds to create more complex testing data. Therefore, creating our own dataset by recording sample keystrokes on one of our own keyboards could allow us to create more sophisticated datasets for testing.

We also consider taking intervals between key presses to identify. In another research paper [2], the authors noticed that an individual tends to type a particular key pair within a similar time interval in most cases, which is influenced by factors such as the arrangement of keys on the keyboard, the anatomy of their hands, and personal typing patterns. Using this, we could further identify a user's typing patterns to better transcribe their keystrokes.

Fourier transform is performed to isolate individual keys from a sampling of mixed keystrokes.

Depending on the software that we will use to record the keystrokes directly from the laptop (ex: Zoom, built-in recorder/Dictaphone) we may need to implement a noise suppression algorithm, or on the contrary levy existing noise cancelling functionalities that prevent some keys from being heard through data augmentation. This may prove to be problematic in zoom for example.

In [1], the authors mentioned their use of a deep learning convolutional network as their model. We may opt for a similar option, as these models are commonly used for tasks such as image and sound recognition.

As an evaluation metric, we will use confusion matrices to evaluate the performance of the classifiers in our experiment. In [1], the authors managed to achieve a 95% accuracy for keystrokes recorded on a nearby phone, and 93% accuracy for keystrokes recorded on a Zoom call. These are the highest percentages achieved. We will attempt to predict the correct classifier with at least 85% accuracy.

**Application :**

A potential application of our project is the sophisticated side-channel attack to listen for passwords through online calls (zooms, for example) or at a distance from another computer. From experience, a Fast API application might provide sufficiently low latency to run our app on a browser. However, we might consider deploying our app through a bundled Python application (with the PyInstaller library) so that the user can run the pre-trained model and get instantaneous results while listening for keystrokes. The latter might also help with its integration into the Zoom app by channelling the audio into our app (with PyAudio for example).

https://github.com/ggerganov/kbd-audio/discussions/31

[1] J. Harrison, E. Toreini, and M. Mehrnezhad, "A Practical Deep Learning-Based Acoustic Side Channel Attack on Keyboa rds," IEEE, 2023/7//, pp. 270-280, doi: 10.1109/eurospw59978.2023.00034. [Online]. Available: http://dx.doi.org/10.1109/EuroSPW59978.2023.00034

[2] R. R. Alireza Taheritajar "Acoustic Side Channel Attack on Keyboards Based on Typing Patterns," *arxiv,* 2024. [Online]. Available: https://arxiv.org/html/2403.08740v1.