UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Karl Raud

Visual Piano Transcription

Master's Thesis (30 ECTS)

Supervisor:
Victor Henrique Cabral Pinheiro, PhD

Tartu 2025

Visual Piano Transcription

Abstract:

Automatic music transcription (AMT) is a field that focuses on extracting symbolic representations from musical performances. Visual piano transcription (VPT) is a subproblem of AMT that uses only visual cues to transcribe piano performances. It is useful in cases where the audio is lost, noisy, or contains multiple instruments. In this work, an end-to-end convolutional deep learning approach for VPT is proposed, which predicts the keypresses of a piano performance, given a video of a person playing it. Three prior researches, including the current state of the art for VPT, were reimplemented under comparable conditions and evaluated against the proposed method on both an existing and a novel, out-of-distribution dataset compiled in the course of this study, to assess whether they can be used in real-world applications. The proposed method is shown to perform well under the tested conditions, surpassing the current state of the art. As a final set of evaluations, the current state of VPT is also directly compared to audio-based piano transcription (APT).

Keywords: deep learning, music transcription, convolutional neural network, computer vision

CERCS: P170 Computer science, numerical analysis, systems, control; P176 Artificial intelligence; H320 Musicology

Visuaalne klaveri transkribeerimine

Lühikokkuvõte:

Automaatne muusika transkribeerimine (AMT) on valdkond, mis keskendub muusikalistest esitustest notatsiooni leidmisele. Visuaalne klaveri transkribeerimine (VKT) on AMT alamvaldkond, mis kasutab klaveriesituste transkribeerimiseks ainult visuaalset teavet. See on kasulik juhtudel, kui soorituse heli on kadunud, mürarikas või sisaldab mitut muusikainstrumenti. Käesolevas töös arendati välja konvolutsiooniline närvivõrk VKT jaoks, mis ennustab klaveri esitusel vajutatud klahvid, lähtudes ainult videost, kus inimene klaverit mängib. Lisaks implementeeriti ka kolm varasemat teadustööd, sealhulgas ka VKT tipptasemel olev töö. Seejärel võrreldi kõiki nelja meetodit võimalikult õiglaselt. Arvestades treenimisandmestiku olemust, hinnati mudeleid nii sarnastes kui ka uudsetes olukordades, et selgitada välja, kas neid saab kasutada reaalmaailma tingimustes. Välja arendatud mudel osutus tõhusaks, saavutades paremad tulemused kui senine tippmeetod. Viimaks võrreldi ka hetkest VKT täpsust helipõhise klaveri transkribeerimise meetodiga.

Võtmesõnad: sügavõpe, muusika transkribeerimine, konvolutsiooniline närvivõrk, masinnägemine

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria); P176 Tehisintellekt; H320 Musikoloogia

Contents

1. Introduction

In the past decade, neural networks have revolutionized the field of machine learning with major advancements compared to hand-crafted approaches for pattern recognition. Deep learning architectures, especially convolutional and recurrent neural networks, have enabled rapid progress in tasks such as image classification [1], speech recognition [2], and natural language understanding [3]. According to a study by LeCun, Bengio, and Hinton [4], the ability of neural networks to automatically learn hierarchical representations from large datasets has led to state-of-the-art results across a wide range of domains, as evidenced by neural network models outperforming human baselines in benchmarks like ImageNet [5] and achieving superhuman accuracy in specific tasks.

Among these advancements, the analysis and transcription of music performances into symbolic representations such as staff notation or MIDI have attracted significant research attention [6, 7]. Automatic music transcription (AMT), and specifically piano transcription, remains a relevant, yet challenging problem, due to the instrument's polyphonic nature: a standard piano contains 88 keys corresponding to notes that can be played simultaneously, leading to complex and overlapping harmonics [6]. Consequently, piano pieces are still not able to be accurately automatically transcribed into sheet music from audio only[1]. Additionally, not all piano performances are captured on digital or MIDI-enabled instruments - the majority of professional and amateur pianos sold globally are acoustic rather than digital[2] which means that musical performances cannot be easily transcribed without modifying the piano.

State-of-the-art systems can achieve transcription accuracies of over 85% for certain audio datasets [8], but these methods almost always require clear and high-quality digital audio as input. Furthermore, in many historical performance archives and educational contexts, only video footage of a performance may exist, either because audio was never recorded, was subsequently lost, or is contaminated by environmental noise (e.g., audience, or other instrument sounds). These real-world scenarios raise the importance of robust visual piano transcription: extracting symbolic musical data from video when audio is unavailable or insufficient.

[1] https://paperswithcode.com/sota/music-transcription-on-maps
[2] https://www.pianobuyer.com/post/acoustic-or-digital-whats-best-for-me

This thesis explores the field of visual piano transcription, reviewing recent approaches, highlighting their successes and limitations, and proposing novel neural network-based methods for accurate extraction of musical information from video data alone. In Section 2 the relevant background information is presented. Section 3 gives an overview of related works. Section 4 details the methodology of our experiments and Section 5 presents the results. Finally, in Section 6 the findings are summarized.

Generative artificial intelligence, specifically ChatGPT[3], was used to improve the readability and clarity of certain paragraphs in this thesis. The use of such tools was limited to text editing and did not contribute to the research content or the formulation of research results.

[3]OpenAI. (2024). ChatGPT [Large language model]. https://chat.openai.com

2. Background

In this chapter, the theoretical background information relevant to our study is presented. We first describe the fundamentals of automatic piano music transcription. Afterwards, different transcription methods are compared and relevant evaluation metrics are defined. Finally, the ResNet architecture is introduced.

2.1 Automatic Piano Music Transcription

Automatic music transcription (AMT) is the process of converting recordings into symbolic representations, such as sheet music or MIDI files. This task is particularly complex for the piano due to its high frequency range and polyphonic nature, where multiple notes are played simultaneously, resulting in overlapping spectra and inharmonic overtones [9].

2.1.1 Piano Overview

A full-sized piano has 88 keys with notes ranging from A0 to C8, as seen in Figure 1. Piano performances consist of a sequence of notes, each having:

- Onset time (When a key is pressed)
- Offset time (When a key is released)
- Pitch (Which key it is)
- Velocity (Loudness of the note)
- Context information, such as pedals that sustain notes after releasing a key

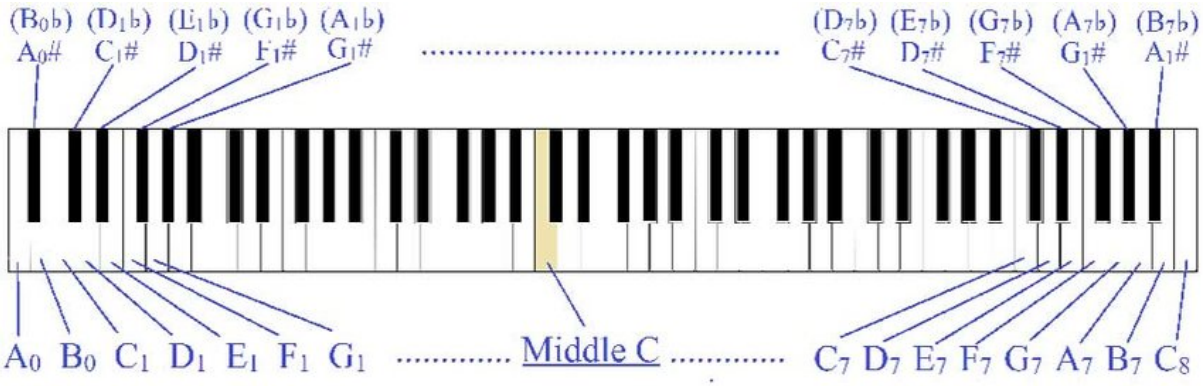

Figure 1. Layout of the piano. [10]

2.1.2 Audio-Based Piano Transcription

Audio piano transcription (APT) systems convert raw audio signals into a time-frequency representation before applying machine learning models to estimate note events. This is frequently done using Mel-Spectrograms (Figure 2). Techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are commonly used to model temporal dynamics and spectral features [8, 11].

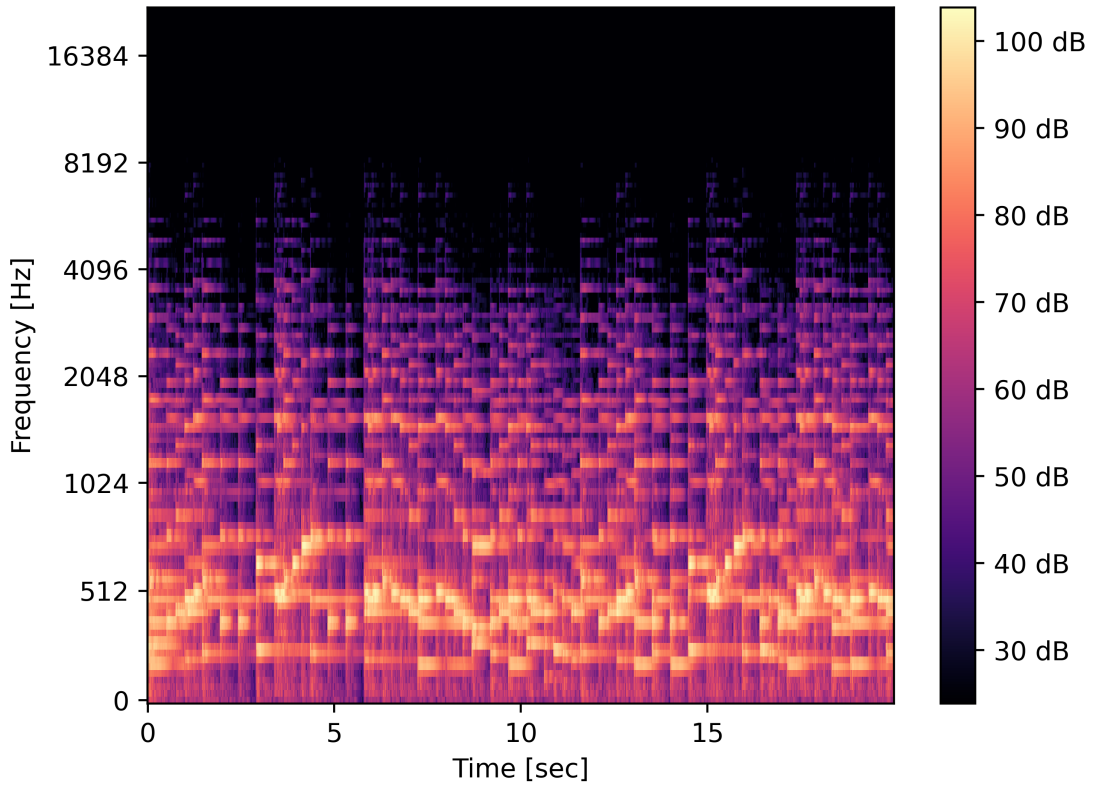

Figure 2. Mel-Spectrogram of the first 20 seconds of Bach - Minuet in G major.

2.1.3 Visual Piano Transcription

Visual piano transcription (VPT) techniques involve capturing video footage of the piano and the performer. Then, the transcription is typically obtained by using a sliding window of size k on the video frames $X_{t-k/2}, ..., X_t, ..., X_{t+k/2}$ to predict notes at time t (Figure 3).

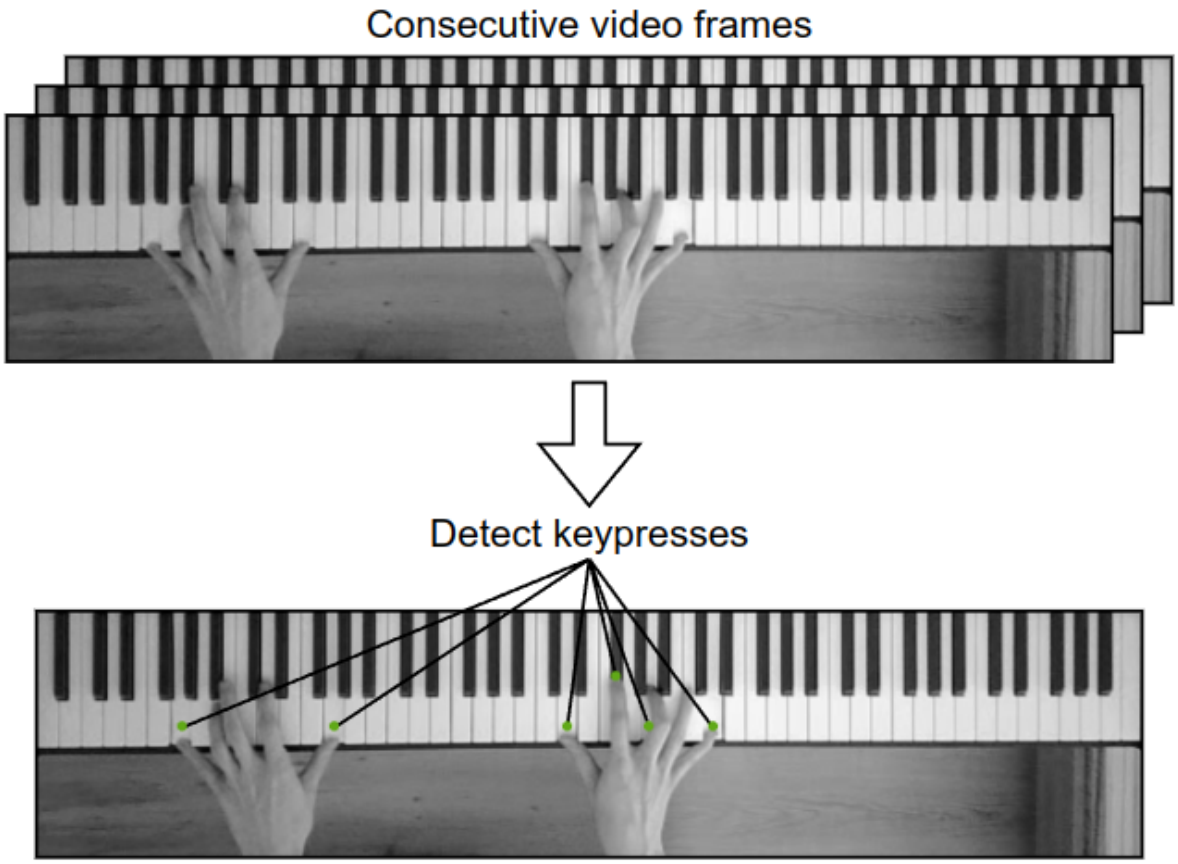

Figure 3. The goal of visual piano transcription. The algorithm takes k consecutive frames for input, and outputs keypresses for the middle frame. For illustration, k=3.

Algorithmical computer vision methods, such as processing frame differences[4] or tracking finger movements[5] have been used to detect piano keypresses. In addition, deep learning models, e.g., convolutional neural networks [12] and transformer-based models [13] have made notable progress in VPT by using multiple consecutive images of the performance to detect notes.

2.2 Comparison of Transcription Methods

Here is an outline of the advantages of APT and VPT. It is worth noticing that modern implementations already combine elements of both.

[4] https://github.com/psuteparuk/PianoKeyDetector
[5] https://github.com/aozkaya/Pressed-Piano-Key-Detection

Strengths of Audio-Based Transcription:

- **Direct Signal Capture:** Provides a direct representation of the acoustic properties of the piano, capturing nuances in dynamics and the sustain pedal.
- **Mature Technology:** More extensive research and established methodologies have led to robust audio transcription systems.
- **Temporal Resolution:** Audio sampling rate is orders of magnitude more frequent compared to video capture, leading to more accurate timing of notes.
- **Robust Sensor Placement:** Physical obstructions, lighting conditions, and the specific location of the microphone do not have a major impact on transcription results.

Strengths of Video-Based Transcription:

- **Direct Key Identification:** Visual cues allow for direct identification of which keys are pressed, potentially reducing the ambiguities present in complex overlapping audio signals.
- **Noise Insensitivity:** Insensitive to background noise, room acoustics, other instruments, and the timbre of the piano.

2.3 Evaluation

An evaluation metric is used to measure the performance of a machine learning algorithm. For VPT, the predictions can be viewed as a fixed collection of events where each event is defined by three variables: fundamental frequency, onset time and offset time [14]. As audio-visual mistimings cannot be perceived in a window of size $\pm$85ms and are acceptable within $\pm$137.5ms [15], note timings are also considered to be correct if within set margins of ground truth.

Precision, which is the fraction of correctly classified instances among all retrieved instances, and recall, the fraction of correctly classified instances retrieved, are defined as:

$$\text{Precision} = \frac{tp}{tp + fp}\,, \tag{1}$$

$$\text{Recall} = \frac{tp}{tp + fn} \tag{2}$$

where tp denotes true positives, fp denotes false positives, and fn denotes false negatives [16]. The F1 score is defined as the harmonic mean of precision and recall:

$$\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{3}$$

2.4 ResNet

He et al. [1] proposed ResNet, a deep learning architecture that has achieved state-of-the-art performance in various classification tasks, such as soil classification [17], music genre recognition [18, 19], and malicious software detection [20]. ResNet is widely adopted due to its ability to address the vanishing gradient problem [21], which often makes training deep neural networks difficult. In traditional architectures, gradients can become extremely small during backpropagation, causing earlier layers to learn very slowly or not at all. ResNet overcomes this challenge by introducing skip connections (also known as residual connections), which allow the input to a layer to bypass one or more layers and be added directly to the output (see Figure 4). Specifically, by adding the input x to the output of the residual function $\mathcal{F}(x)$, ResNet enables gradients to flow more easily through the network during backpropagation, alleviating the vanishing gradient problem.

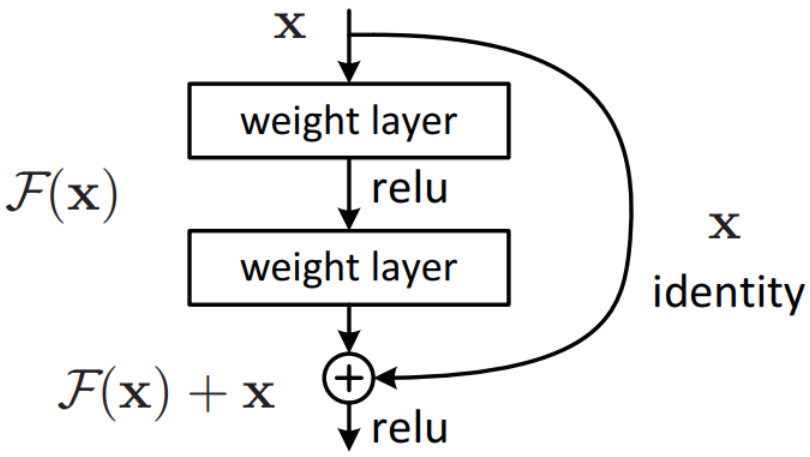

Figure 4. A single building block of ResNet [1].

3. Related Works

In this chapter, the main papers implemented and considered for direct comparison with the thesis' approach will be briefly explained. It is worth noticing that, as far as could be determined from the reviewed literature, the paper mentioned in Section 3.3 is the current state of the art for VPT. Finally, Section 3.4 describes an audio transcription approach which was used for comparison between audio and video transcription.

3.1 Sight to Sound: an End-to-End Approach for Visual Piano Transcription

Koepke et al. [12] proposed a convolutional neural network for visual piano transcription. Their proposed model, as seen in Figure 5, is a modified ResNet-18 [1] architecture. After forwarding five consecutive input frames separately through the first block of ResNet, the **aggregation module** concatenates the features using a single 3D convolution, capturing semantic differences over time. After the third block of ResNet, the **slope module** is introduced, which augments the feature map with additional positional information for the model to detect the correct key.

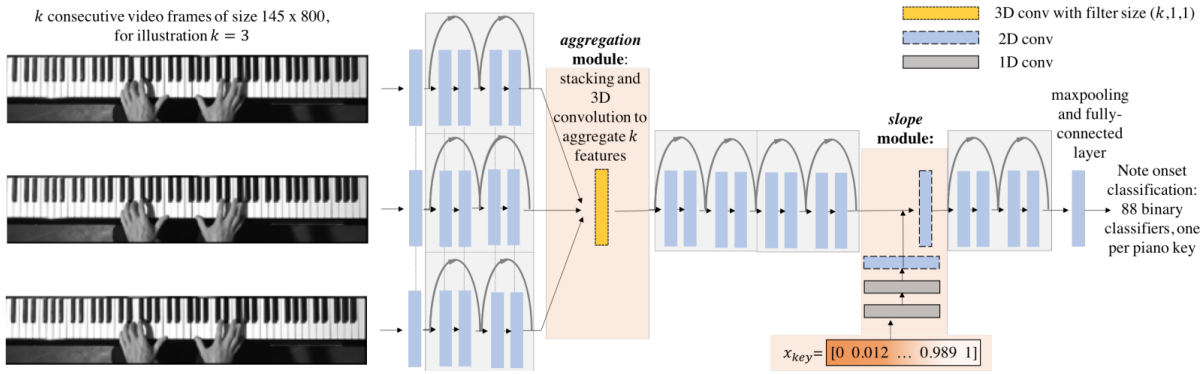

Figure 5. An overview of Koepke et al. network architecture [12].

The authors demonstrated that, by combining the predictions of both audio and visual transcription models, the results would improve compared to only using audio information.

3.1.1 PianoYT dataset

Koepke et al. also created a VPT dataset consisting of 228 YouTube videos and about 20 hours of piano performances. The dataset is publicly available on their project page[6]. The pseudo

[6] https://www.robots.ox.ac.uk/~vgg/research/sighttosound/

ground truth was created using the Onsets and Frames [8] audio transcription framework. The coordinates of the bounding box of the piano keyboard are also given for each of these videos.

3.2 Audeo: Audio Generation for a Silent Performance Video

Su et al. [22] introduced a VPT pipeline for predicting the audio for a piano performance video. Their pipeline consists of three modules (Figure 6):

- **Video2Roll Net** extracts note onsets and offsets from consecutive frames of the video. This is further detailed in Section 3.2.1.
- **Roll2Midi Net** is a generative adversarial network that refines the note predictions by using a much larger time window (4 seconds) compared to Video2Roll Net. Its goal is to mitigate the inaccuracies of longer notes that are physically pressed, but for which the sound has diminished over time.
- **Midi Synth** is a PerfNet [23] model that synthesizes audio for the generated notes.

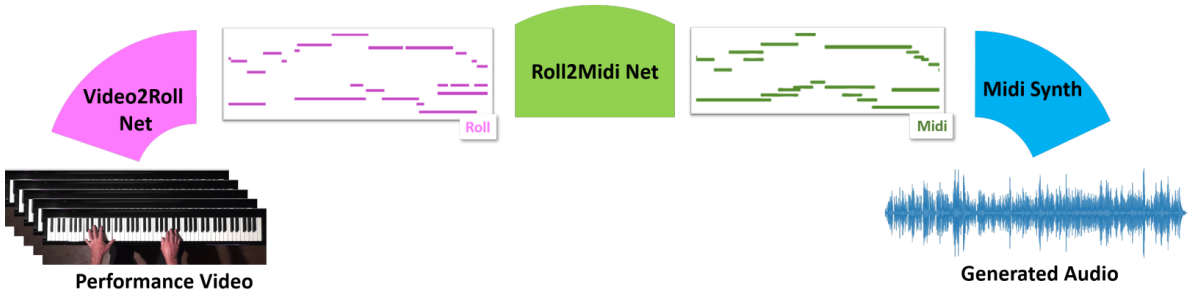

Figure 6. An overview of Su et al. Audeo pipeline [22].

3.2.1 Video2Roll Net

The Video2Roll Net (Figure 7) consists of a ResNet-18 [1] backbone, feature transform, feature refinement, and correlation learning modules. The feature transform and feature refinement module form a multi-scale feature pyramid network, which detects smaller visual cues of the piano keys. The correlation learning module learns spatial dependencies and semantic relevance of the features to detect the correct key.

3.3 A Transformer-Based Visual Piano Transcription Algorithm

Zivanovic et al. [13] propose a fully transformer-based pipeline for visual piano transcription based on a pretrained VideoMAE [24] backbone for onset and pitch detection. In contrast to previous works, they use a lower spatial resolution of 224×224 but cover a longer temporal window (16 frames).

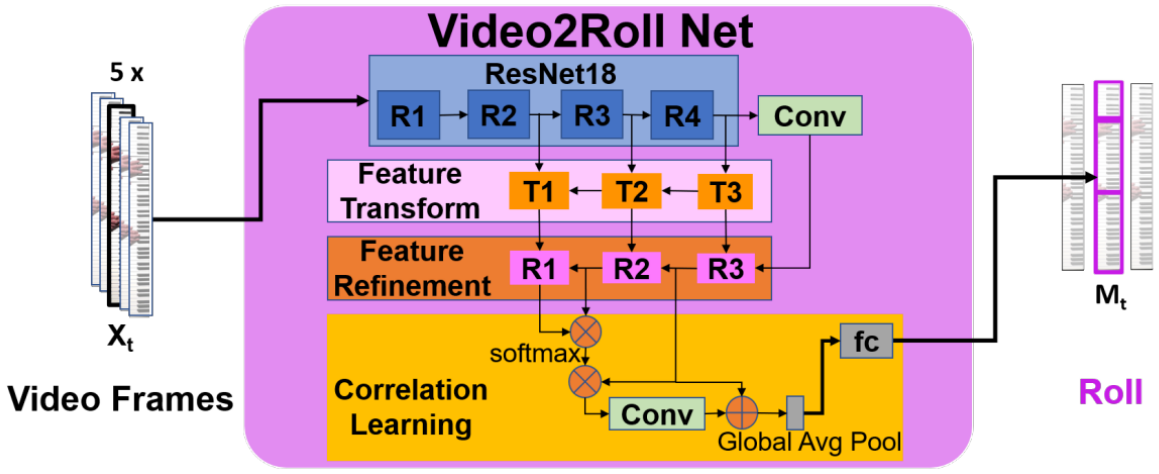

Figure 7. An overview of Su et al. Audeo Video2Roll Net architecture [22].

To support automatic and robust piano localization, they also train a YOLOv8 [25] model to detect piano keyboards and generate crops, removing the need for fixed or manually defined regions. This allows their pipeline to handle more challenging and realistic recording scenarios.

A key focus is on image preprocessing: how to best convert the raw crops into the model's required square input, given the low resolution. They experiment with several strategies, such as stretching, aspect-ratio normalization and splitting/stitching crops (see Figure 8).

For training and evaluation, the authors used a new (but not public) R3s dataset and the PianoYT dataset. Their end-to-end approach delivers the strongest published results for visual onset detection to date, with F1-scores of 83.81 on R3s and 67.31 on PianoYT, outperforming previous CNN-based systems and demonstrating how transformers benefit from longer temporal context and robust data processing.

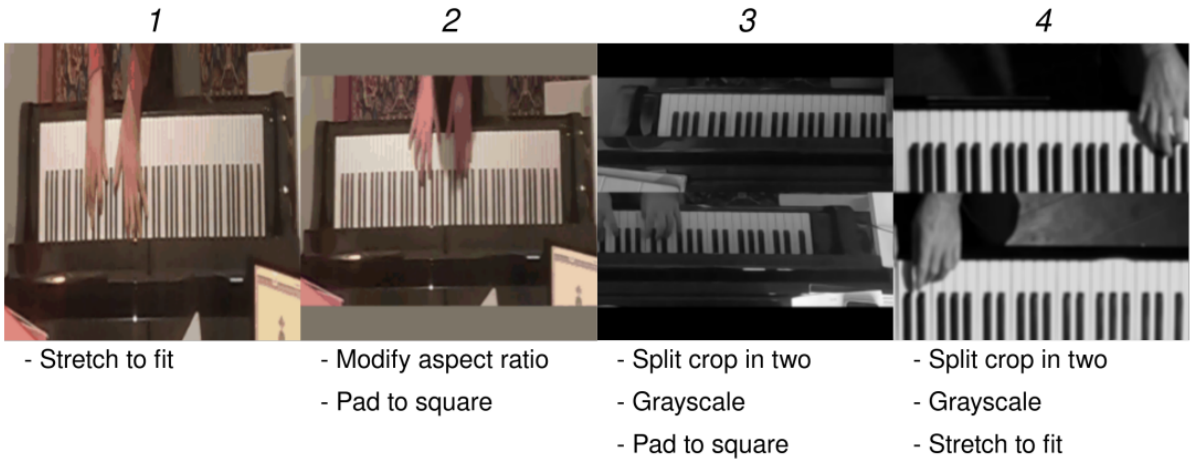

Figure 8. Different image preprocessing techniques explored by Zivanovic et al. [13].

3.4 A Data-Driven Analysis of Robust Automatic Piano Transcription

Kong et al. [26] proposed a framework for piano transcription from audio. They use a deep neural network architecture that combines convolutional layers for feature extraction from input log-mel spectrograms (see Figure 2) with gated recurrent units to capture temporal dependencies. Ultimately, they extract the onsets, offsets, velocities and the pedalling information for the piano performance.

However, such models are known to often learn to overfit to the specific acoustic conditions of their training data, making them less robust to recordings from different pianos or rooms. To address this, Edwards et al. [27] focused on making the training process more robust and generalizable, rather than altering the network architecture itself. The authors created a new, professionally studio-recorded version of the MAESTRO dataset [28] and used various audio augmentations during training, such as random equalization, background noise, pitch shifting, and reverb. By increasing both the diversity of training data and applying these augmentations, they achieved new state-of-the-art performance on the MAPS dataset [29] (88.4 onset F1-score) without using any of its training data, significantly improving out-of-distribution generalization. This demonstrates that a data-centric approach, with diverse training data and effective augmentation, is essential for building more generalizable music transcription systems.

4. Methodology

In this chapter, the methods and approaches of this study are detailed, including the dataset, evaluation, model architecture, and training of models.

4.1 Dataset

For all experiments, the models are trained on the PianoYT dataset (described in Section 3.1.1). The dataset consists of videos and midi files.

An example Midi file looks like this:

```
[Metadata]: ticks_per_beat=220
MetaMessage('set_tempo', tempo=500000, time=0)
event: note_on channel=0 note=45 velocity=39 time=4534
event: note_on channel=0 note=65 velocity=58 time=0
event: note_on channel=0 note=45 velocity=0 time=84
event: note_on channel=0 note=52 velocity=48 time=57
event: note_on channel=0 note=67 velocity=56 time=56
event: note_on channel=0 note=53 velocity=46 time=14
event: note_on channel=0 note=65 velocity=0 time=14
...
```

Tempo describes the amount of microseconds per quarter note. Ticks_per_beat denotes the amount of ticks per quarter note. The time of the i-th event Δt_i denotes the variation in time from the previous event in ticks. Thus, the **timestamp** T in seconds for each event can be calculated as:

$$T_i = \frac{\sum_{j=0}^{i} \Delta t_j \times \text{tempo}}{\text{ticks\_per\_beat} \times 10^6} \tag{4}$$

Event velocity denotes the loudness of the note. It was discovered that a velocity value of > 0 represents **onsets**, and a velocity value of 0 denotes **offsets** (see Section 2.1.1). The event note variable is an integer that describes the **pitch**. The note indices of an 88-key piano (Figure 1) range from 21 (A0) to 108 (C8).

Each midi event is assigned to the nearest frame in the piano performance video such that $|T_i - \text{frame\_timestamp}| \leq \frac{1}{2 \times \text{fps}}$. The timings of the midi events in the PianoYT dataset were consistently approximately 2 video frames late, so an offset of $\frac{-2}{\text{fps}}$ seconds was added to all timestamps for the midi events.

4.1.1 PianoVal dataset

To evaluate the generalizability of the models to unseen data and new environments, an additional validation dataset, ”PianoVal”, was constructed for this study. It consists of 19 minutes of amateur piano performances captured as 30fps video at a resolution of 640x480. The dataset consists of two videos of scale exercises, one video of 4 chromatic scales in each octave (where every note of the piano is used four times) and 7 different musical performances in various keys (a key of a piano piece describes which notes are most frequently used). The piano used was a Casio CDP-S110 digital piano which was connected using a USB connection that allowed for accurate extraction of the ground-truth midi events by reading the USB data stream with aseqdump[7]. The video piano frame crop coordinates were manually set so that the input images seem similar to the ones provided in the PianoYT dataset. A comparison of sample images from each dataset is shown in Figure 9.

Figure 9. Images from the PianoYT dataset (top) and the PianoVal dataset (bottom).

4.2 Evaluation

The models output logits (probabilities) for each frame and note, with shape *[video_frames, 88]*. These predictions are postprocessed until binary classifications are obtained. Note onset times are calculated from the video fps and the index of the predicted frame. Note frequencies are computed using the formula[8]:

$$F = \frac{440}{16} \times 2^{\text{note\_idx}/12} \tag{5}$$

Where note_idx is in the range of $0 - 87$ describing which note it is on a 88-key piano.

[7] https://man.archlinux.org/man/extra/alsa-utils/aseqdump.1.en

[8] https://www.ece.iastate.edu/~alexs/classes/2016_Spring_575/HW/HW5/files/piano-key-freq-wikipedia.pdf

For evaluation, mir_eval [30] was used, which implements the music information retrieval (mir) evaluation metric proposed by Bay et al. [14] From note onset times and fundamental frequencies, the precision, recall, and F1-scores are computed.

Previous works have used various onset mistiming thresholds such as ±50ms [22] and ±100ms [13]. In this work, a threshold of ±100ms is used for all experiments. As described in Section 2.3, this threshold is in the middle of the noticeability and acceptability ranges.

4.3 Proposed model

We propose a novel model architecture for piano onset detection. In the following paragraphs, we will discuss the motivation for our choices. Afterwards, we describe the proposed model in detail.

CNN vs Self-attention. Although Zivanovic et al. [13] achieved state-of-the-art performance on the PianoYT dataset using a transformer model, they still suffer from multiple flaws, compared to CNNs. The massive number of parameters and the nature of the attention mechanism cause attention-based models to generally perform slower than their convolutional counterparts [31]. Furthermore, because of their weaker inductive bias, it is known that transformers tend to require much larger training datasets in order to generalize on the training data [32, 33]. Due to these reasons, the convolutional neural network architecture was chosen.

The nature of the piano. As pianos are rectangular in shape and are mostly horizontally placed on an image, an assumption can be made that image's horizontal information carries information about the state and the location of every note. In contrast, vertical information only carries information about a single note. Thus, preserving image width throughout the network is more valuable while using strided convolutions.

Importance of resolution. It is observed that the visual differences of pressed or unpressed piano keys are minor. On small image resolutions such as 300x300, only pixel-level changes can be observed. This means that strided convolutions should be used with caution, by not eliminating the smaller details of the image in the first layers.

Temporal window. The number of consecutive frames given to the model was chosen to be 15 frames. This is different from many previous works that have used a smaller temporal window of 5 frames [12, 22]. Thus, considering a 30fps video, each input sample corresponds to 0.5s of video. By using a larger temporal window, it is more probable that the input image sequence will contain both pressed and unpressed instances of each key. This means that the model can

more frequently use frame differences to detect events. In addition, a larger temporal window can mitigate misidentifying onsets from offsets if both occur within the temporal window.

4.3.1 Model architecture

The model consists of **R-Blocks**, which are a modified version of the original ResNet building blocks (Figure 4). The ReLU layers were replaced with LeakyReLU to counteract the ”dying ReLU problem” [34, 35]. The input is first passed through a 1×1 convolution layer that outputs a representation with the target dimensionality. After which it is fed into a $n \times n$ convolution which is optionally strided. The residual skip connection consists of a single 1×1 convolution, which matches the output dimensionality of the left-side branch in Figure 10.

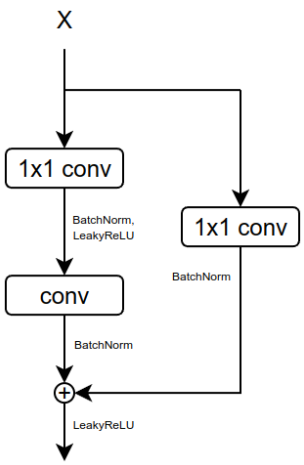

Figure 10. A single R-Block.

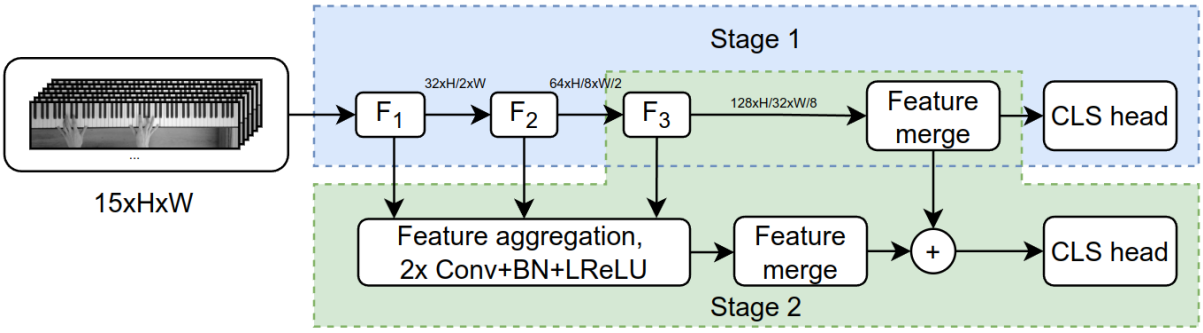

Figure 11. The architecture of the proposed model.

Figure 11 depicts the architecture of the model. It has three **feature extraction modules** F_1, F_2, F_3 which consist of 5, 6, and 8 sequential R-Blocks respectively. Note that throughout feature extraction, horizontally strided convolutions are used less in the earlier stages to preserve image detail.

The **feature aggregation module** is designed to extract accurate note timings by reintroducing the lower-level features, such as frame differences, to the higher-level features such as locations of each note. It uses bilinear interpolation to resize the F_1 and F_2 feature maps to the shape of the output of F_3, after which, all three are concatenated channel-wise. Then, the combined features are forwarded through two Conv-BatchNorm-LeakyReLU blocks.

The **feature merge module** eliminates the height and width dimensions using a fully-connected layer with shared weights channel-wise. This method was tested to outperform the analogous 2DAdaptiveAveragePooling operation in the original implementation of ResNet.

The **classification head** is a fully-connected layer that is followed by a sigmoid activation function to predict the note onsets of shape *[88]*.

4.3.2 Training

The model is trained in two stages (see Figure 11). The first stage is designed for the model to learn to detect the correct pitch and the approximate onset time. The second stage is designed to refine the note timings.

During training, similarly to Zivanovic et al. [13], positive labels were duplicated to nearby frames (note span in Table 1) to increase the amount of positive samples in the dataset. Preliminary experimentation concluded that without duplicating notes over multiple samples, the model would overfit by memorizing the exact frames where the keys are pressed due to the low amount of samples. During stage 2 training, the notes are not span over multiple frames, but instead, the first two feature extraction modules are frozen so the model cannot learn to memorize pixel-level features. In addition, a percentage of negative samples (without any notes) was randomly removed to equalize the class balance. The described data preprocessing is shown in Figure 12.

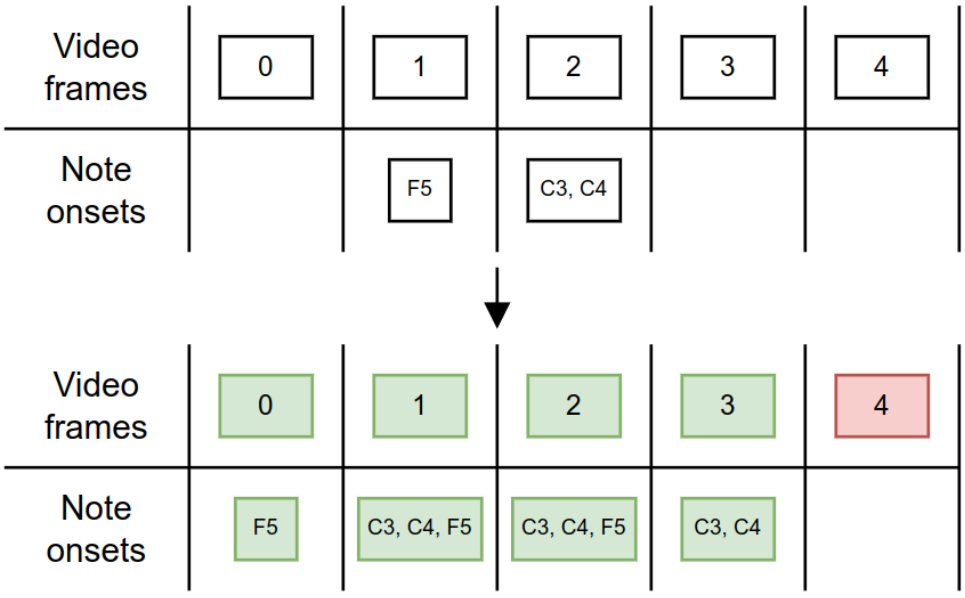

Figure 12. Illustration of increasing the note span of the training dataset. Upper image is the original data. Lower image shows samples after applying note spanning with radius 1. A percentage of negative (red) samples are randomly removed from the dataset.

Table 1. Differences between training stages.

	Stage 1	Stage 2
Percentage of discarded samples without notes	95%	80%
Positive sample weight	3	9
Positive sample span radius	1	0
Training epochs	4	3
Frozen layers	-	F_1, F_2
Unused layers	Feature aggregation Stage 2 feature merge Stage 2 classification head	Stage 1 classification head

Weighted binary cross-entropy loss was used to optimize for each of the 88 classes. AdamW optimizer was used, with a learning rate of 0.001 and L2 regularization factor of 0.001. Learning rate was scheduled using cosine annealing and was reset in the start of stage 2. The model was trained for a total of 7 epochs with a batch size of 64.

All of the videos were transcoded to a frame rate of 30 frames per second using FFmpeg [36]. For data augmentations, unless otherwise specified, the default parameters of Albumentations python library [37] were used for the following transformations:

- Grayscale
- Resize to a width of 640 and a height of 120 pixels
- Spatial jitter of x (-5 to 5) and y (-20 to 10) pixels
- Image brightness jitter of 50% to 150% of the original value
- Gaussian noise with a standard deviation range of 0.05 to 0.1
- Sharpening of the image with a probability of 0.2
- Random rotation up to ± 5 degrees

During inference, a threshold of 0.7 was used to binarize the predictions. Whenever an onset for the same key was detected over multiple consecutive frames, only the middle frame of the sequence is set to true.

4.4 Implementation details

For reproducibility, the implementation specificities of related works are detailed here. All related works were implemented as faithfully as possible to ensure fair comparison of model architectures. Hence, unless otherwise stated in this section, implementations can be assumed to follow the descriptions provided in the respective papers discussed in Section 3.

4.4.1 Sight to Sound

As the authors have not provided the official implementation for their ”ResNet+aggregation+slope” network, their model architecture is reproduced as follows:

For the ResNet18 [1] backbone, pre-trained weights on ImageNet [5] were used.

For the implementation of the slope module, the two 1D convolution channel sizes were chosen so that both output a feature map of size *[B, 64, 88]*. As the desired output feature amount of the slope module $64 \cdot 10 \cdot 50 \neq 0 \mod 88$, the convoluted slope vector is interpolated to size *[B, 64, 1, 100]* after which it is spatially cloned to *[B, 64, 5, 100]* and reshaped to *[B, 64, 10, 50]* to match the output shape of the third ResNet block.

Additionally, the figure (Figure 5) provided in the paper included a maxpooling operation after the fourth ResNet block, but avgpooling was used as in the original implementation of ResNet [1].

4.4.2 Audeo

The Audeo Video2Roll Net was modified so that it predicts all 88 keys of the piano, instead of 51. For class balancing, random sampling was used, which balanced equally all instances of onsets and whether no keys are pressed.

As the model was initially tuned for predicting note onsets and offsets, the provided loss function weighted the classes equally and the model performed poorly due to the low amount of positive samples. To counteract this, a weight of 3 was assigned to positive samples, and during training, the ground truth notes were span over 3 frames to increase the amount of positive samples.

4.4.3 A Transformer-Based Visual Piano Transcription Algorithm

The preprocessing technique of Figure 8 image 4 was used as recommended by the authors. For video augmentations, vertical spatial jitter of −20 to +5 pixels and color jitter with the default values of Albumentations python library [37] were used.

4.5 Additional experiments

The following supplementary experiments are reported for scientific thoroughness; although methodologically sound, their results fell short of state-of-the-art performance. For the validated experiments and results, the reader may want to skip directly to Section 5.

4.5.1 Foundational models as feature extractors

Foundational models such as SAM2 [38] or DINOv2 [39] achieve excellent object detection and segmentation results without any additional training. As these generic models can extract useful semantic features for almost any task, the CLIPSeg [40] prompted segmentation model was used to extract logit masks for ”hand” and ”piano keys” prompts. An example is shown in Figure 13. These logits were concatenated to the model input to provide additional contextual information about the location of the pianist's hands and the piano. Unfortunately, this did not yield measurable improvements in the predictions.

Figure 13. Image from the PianoYT dataset along with CLIPSeg [40] logits. Original image (top) and the upscaled masks of ”hand” (middle) and ”piano keys” (bottom).

4.5.2 Temporal super-resolution

As VPT methods are frequently trained to predict onsets for each frame, their temporal accuracy is tied to the framerate (fps) of the video. To counteract this, an additional U-net [41] was trained that processes n seconds of VPT model prediction logits of shape $[n \times fps, 88]$ and outputs predictions with 4 times higher temporal accuracy of shape $[4 \times n \times fps, 88]$. Experiments concluded that the results did not improve due to the fact that the logits did not contain enough

sub-frame timing information to achieve accurate super-resolution. For future research, a super-resolution module could be incorporated directly into the VPT model, where sub-frame timings can potentially be extracted from the initial images, not the outputted logits.

4.5.3 Training without note ground truth

Most VPT works have involved training a model that predicts note pitches and onsets, by using corresponding ground truth note pitches and onsets. Note events are often extracted using an audio transcription model and the resulting pseudo ground-truth is not always accurate. An alternative approach was attempted, which set the audio signal as ground truth instead. This method allows for training on any recording of piano performance that contains the corresponding audio.

The pipeline used was as follows: first predict note onsets from the input video stream. Then transform the predicted notes into the predicted audio signal. From the video stream, also extract the corresponding ground-truth audio, at a predefined sample rate and length. Finally, use a frequency-based audio loss, such as STFT loss [42] to update the weights of the entire pipeline by comparing the predicted audio to the ground-truth audio. For simplicity, this pipeline now is notated as ”video→notes→audio”. During model inference, the note pitches and onsets can be extracted from just the video→notes model.

For the video→notes model, the proposed model in Section 4.3.1 was used. For the notes→audio model, multiple methods were tested, but each had a fundamental flaw:

- From the predicted note pitches and loudnesses, the corresponding sinusoidal pitch waves were computed. Every individual sine wave was merged into a multi-frequency single-channel audio signal. All of it was implemented in a differentiable manner, where gradients could backpropagate through the audio→notes part of the pipeline. Unfortunately, this method outputted unrealistic audio which did not accurately capture the timbre and background noise of the piano audio. Thus, the gradients that backpropagated through the notes→video part of the pipeline contained too much noise.
- For the second approach, from the predicted note pitches and loudnesses, a predefined audio signal was outputted as the predicted audio. Samples of prerecorded audios from

a piano soundfont[9] were used. This method introduced a problem where the indexing operation of the sample was not differentiable. By exploiting the fact that the samples can be sorted by their fundamental frequency and loudnesses, a novel deep-learning 2D binary search optimization method was developed, which attempted to differentiate the indexing operation. The model predicted indexes of the note pitch $i \in 0..87$ and note loudness $j \in 0..127$ in the soundfont sample space S_{ij}. Then, a weighed sum of samples $\sum_{a\in\{0.5,1,2\},b\in\{0.5,1,2\}} w_{a,b} \times S_{a\times i,b\times j}$ was outputted. The weights w are a combination of the fractional part of predicted i and j. The results of this method indicated, that for a smaller search space of $i \times j \approx 1000$, it successfully optimized for indices i and j, but for the whole search space, the optimization did not converge.

- Finally, a differentiable digital signal processing (DDSP) model proposed by Engel et al. [43] was tested for the notes→audio model. Although it had the most promising results, the pipeline did not converge even when using a simple single-note dataset. It is hypothesized that the vanishing gradients problem [21] is the cause. This pipeline is illustrated in Figure 14.

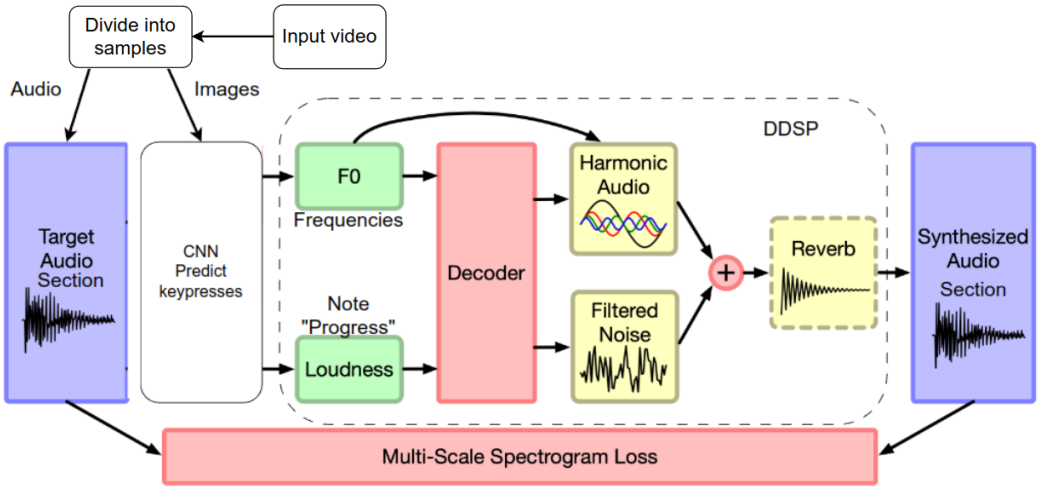

Figure 14. Video→notes→audio model training pipeline using DDSP [43].

[9] https://github.com/sfzinstruments/SalamanderGrandPiano

5. Results and Discussions

This chapter describes the outcomes of the validated experiments conducted and the discussions put forward.

5.1 Numerical comparison of models

This section compares the proposed model against the implementations introduced in Section 3. For simplicity, in Sections 5.1 and 5.2, these approaches are notated as STS (Sight to Sound [12]), Audeo (Audeo Video2Roll Net [22]) and transformer-based (A Transformer-Based Visual Piano Transcription Algorithm [13]). On the tables, unless stated otherwise, the bolded numbers indicate the best comparative results.

5.1.1 Training and inference

For training all models, a single Nvidia A100 GPU was used, provided by the University of Tartu Rocket HPC cluster[10]. The inference speed is measured on a consumer-grade RTX 4060TI GPU because high performance data center graphics accelerators such as Nvidia A100 are probably not as accessible for real-world applications.

Table 2. Comparison of model performance, training time, parameter count and inference speed.

Model	Performance (GFLOPS)	Training time (hours)	Parameter count	Inference speed (frames per second)
Transformer-based [13]	135.24	33.26	86.295M	18.4
Sight to Sound [12]	9.63	19.3	11.986M	210.2
Audeo [22]	**5.24**	**10**	12.718M	**413.4**
Proposed model	9.77	10.52	**1.693M**	108.7

Table 2 summarizes the inference speeds of the models. The transformer-based method requires the most computational resources due to the large size of the VideoMAE [24] model. It is the only approach that cannot be run in real time on 30fps video, on which the model was trained.

For real-time applications – for example, as a script where the detected notes are played back using pre-recorded piano sounds[11] – it should be noted that every model introduces additional latency

[10] https://hpc.ut.ee/services/HPC-services/Rocket

[11] https://medium.com/@karlraud11/visual-piano-note-detection-96af5914ec0b

due to their reliance on following frames to predict onsets for the current frame. Accounting for inference speed and the temporal window size, the estimated onset latencies for each method are calculated using the following formula:

$$\text{latency} = \left\lceil \frac{\text{window\_size}}{2} \right\rceil \div \min\left(\text{inference\_fps}, \text{video\_fps}\right) \tag{6}$$

In case of real-time applications, the estimated latencies are as follows:

- Transformer-based: 435ms
- STS: 100ms (during the training of this model, the input video framerate was not fixed, so 30 was chosen as a common value)
- Audeo: 120ms
- Proposed model: 267ms

When training these models to predict onsets for the last frame of the input sequence instead, these latencies can be significantly lowered to $\frac{1000}{\text{inference_fps}}$ms.

5.1.2 PianoYT test set

Table 4 shows the precision, recall and F1-score for each of the approaches on the PianoYT test set. The proposed model with only stage 1 training (see Section 4.3.2) was also evaluated. For STS and transformer-based approaches, the achieved results are not the same as described in their respective papers.

Table 3. Precision, recall and F1-score on the PianoYT test set.

Model	Precision	Recall	F1-score
Transformer-based [13]	77.43	70.40	73.74
Sight to Sound [12]	74.89	60.69	67.05
Audeo [22]	71.38	62.60	66.70
Proposed model stage 1	79.80	71.18	75.24
Proposed model	**81.70**	**71.77**	**76.41**

Koepke et al. [12] reported a precision of 62.23 and a recall of 73.00, which are different from our reimplementation of 74.89 and 60.69 respectively. Due to the fact that the F1-scores are close (66.63 and 67.05), it is hypothesized that balancing the class weights for the loss is implemented in a slightly different manner, which caused the precision and recall metrics to be flipped.

Zivanovic et al. [13] reported a precision and recall of 69.84 and 65.59 which are lower than our reimplementations: 77.43 and 70.40. This could be due to the fact that, in this work, all note events in the PianoYT dataset have a timing offset of 2 frames (described in the end of Section 4.1). During preliminary testing, adding this offset improved the results of the proposed model as well, indicating that the midi files provided by Koepke et al. [12] are indeed unsynchronized with regards to the YouTube videos.

5.1.3 PianoVal dataset

To test the models generalizability to new datasets, the models are evaluated on the PianoVal dataset (see Section 4.1.1). In addition, the audio transcription method introduced in Section 3.4 is evaluated as well. Table 4 gives an overview of precision, recall and F1-score on the PianoVal dataset.

Table 4. Precision, recall and F1-score on the PianoVal dataset.

Model	Precision	Recall	F1-score
Transformer-based [13]	**85.20**	59.98	70.40
Sight to Sound [12]	23.00	35.91	28.04
Audeo [22]	33.95	6.70	11.24
Proposed model stage 1	71.55	78.49	74.86
Proposed model	75.99	**79.72**	**77.81**
Edwards et al. [27]	**86.19**	**88.04**	**87.11**

To simulate real-world behavior, all of the methods were evaluated with the same parameters as with the PianoYT test set. This includes thresholds, data preprocessing and postprocessing. No midi event timing offsets were used due to the millisecond-scale accuracy of the PianoVal ground-truth events.

Additionally, Figure 15 gives and overview about separate key F1-scores for the proposed and the transformer-based models. It is important to understand that all keys are not represented equally in this dataset, but also that all keys are represented at least 4 times. The exact key counts are given in Figure 16. Thus, the F1-scores for notes with very low counts are to be taken with a grain of salt.

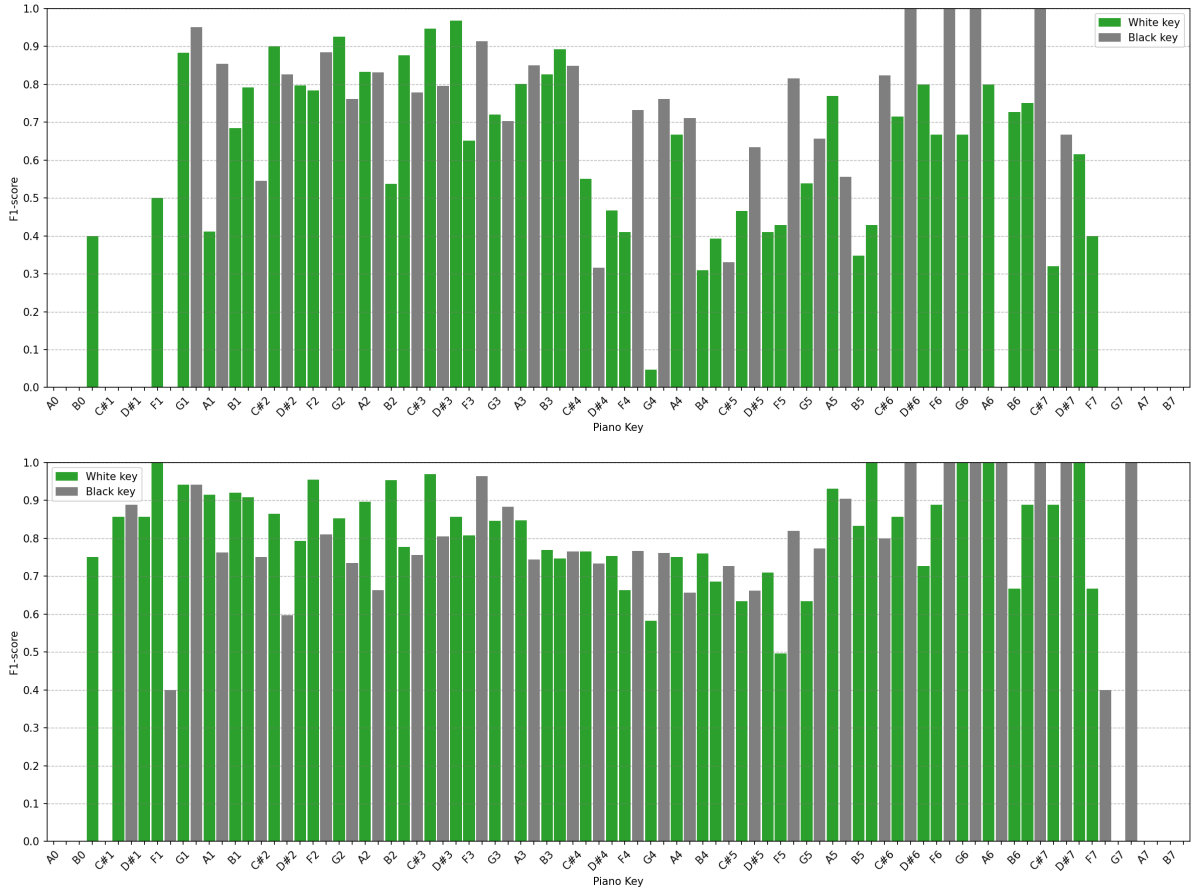

Figure 15. F1-scores for each piano key on the PianoVal dataset. Black keys of the piano are colored gray and white keys green. Predictions by transformer-based model (top) and the proposed model (bottom).

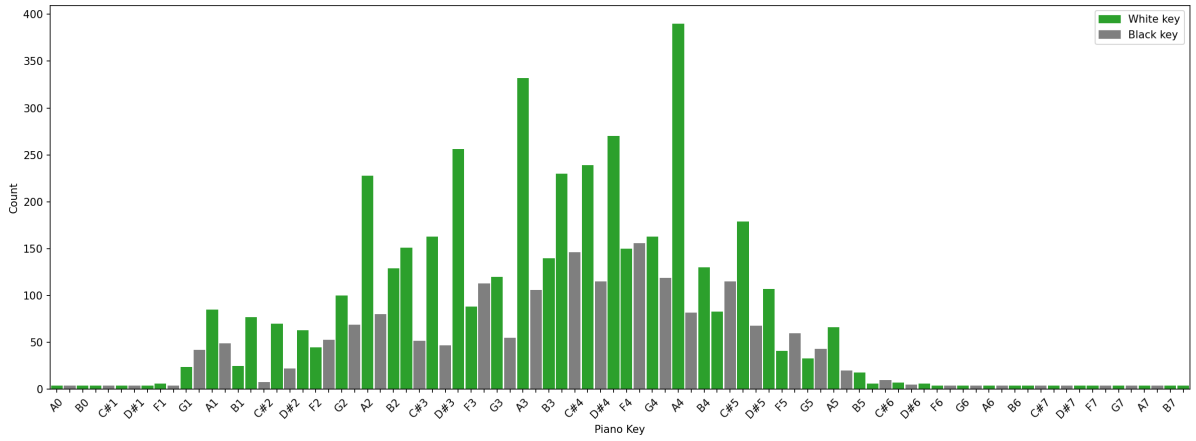

Figure 16. Count of different notes in the PianoVal dataset. Black keys of the piano are colored gray and white keys green.

5.2 Discussion

In this section, the results presented in Section 5.1 are discussed and possible reasonings for them are provided.

5.2.1 Multi-stage training

From Tables 3 and 4, it can be seen that training the proposed model only with stage 1 already achieved great results. Lowering the note span of the dataset (Table 1) and fine-tuning the model with the feature aggregation module of stage 2 did not affect recall much, but improved the precision noticeably. This means that roughly the same amount of onsets were detected for stage 1 and stage 2 models, but the stage 2 model found the exact timing of the notes more accurately. Thus, the hypothesis stated in Section 4.3.2, that stage 2 learns to predict accurate note timings is true.

5.2.2 Generalizability to unseen data

Analyzing Table 4, it becomes clear that STS and Audeo failed to generalize the onset prediction task from the PianoYT training data to the PianoVal validation data. Possible reasons could include the different resolution of the videos (1920x1080 for PianoYT and 640x480 for PianoVal) or the training hyperparameters.

Another crucial factor influencing the generalization ability of the models is the parameter count (shown in Table 2). The proposed model has only approximately 1.7 million parameters, in contrast to STS and Audeo, both with around 12 million parameters. Too high parameter counts can cause the model to overfit, by memorizing too many details from the training set, and thus, perform orders of magnitude worse on the validation data [44]. Although the transformer-based model had an even larger parameter count, at about 86 million, models using attention mechanisms have been shown to scale well with high parameter counts [45].

5.2.3 Input image resolution

The image resolutions of the proposed model (640x120), Audeo (900x100), and STS (800x145) are all much higher than the transformer-based approach (224x224). More precisely, due to the usage of the preprocessing method described in Figure 8 (rightmost image), the effective resolution of the keyboard can actually be described as 448x112. Representing all of the 88 keys of the piano with 448 horizontal pixels means that only about $\frac{448}{88} \approx 5$ pixels describe the state of each key. This is one potential reason on why the proposed model outperforms the state-of-the-art transformer-based approach.

Additionally, STS and Audeo models both include the same first 4 operations at the start of the network: convolution with stride 2, batch normalization, ReLU operation, and a maxpool operation with stride 2. This means that, at the start of the network, the resolution of the image is divided by 4. This is also one of the reasons why the inference on these models is faster (Table 2). Although the loss of detail is somewhat mitigated by increasing the amount of feature channels throughout these operations and the size of the convolution kernel, the authors of this work hypothesize that the usage of strided convolutions is the primary reason why the proposed model outperforms these methods in regards to prediction accuracy. By using more convolution operations on the full-sized image at the start of the neural network, more semantic features can be extracted.

5.2.4 Note-level metrics

The note-level F1-scores are depicted in Figure 15. The note-level metrics were colored so that black and white keys of the piano are colored distinctively. It can be observed that there are no obvious differences in the F1-scores of black and white keys for each model. Both models failed to predict some notes on the edges of the piano. From Figure 17 it may be noted that some of these notes are indeed simply poorly represented in the PianoYT training dataset.

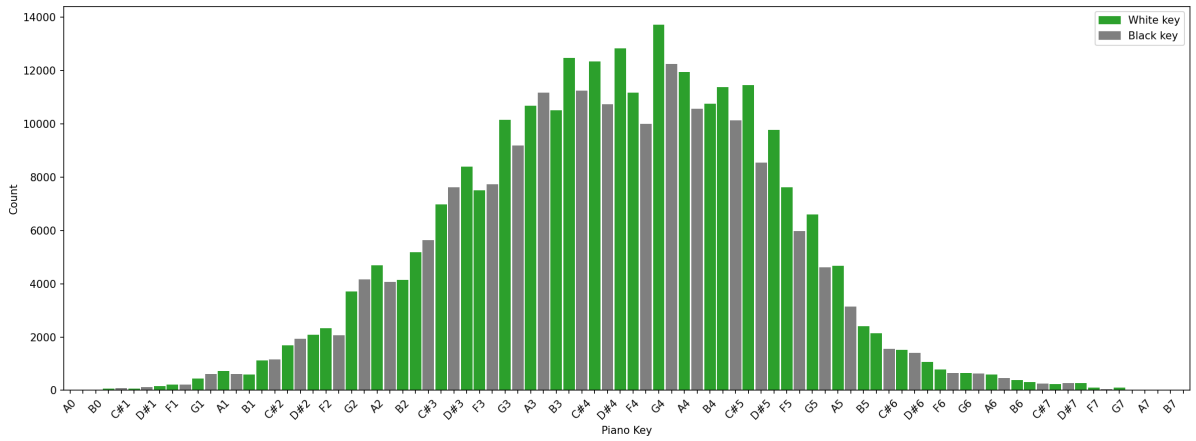

Figure 17. Count of different notes in the PianoYT training dataset. Black keys of the piano are colored gray and white keys green.

Additionally, piano performances frequently consist of simpler left-hand chords (harmony) and more complex right-hand sequences (melody) [46]. As the melody of a musical piece stands out as more influential to the listener [47], it is more important to transcribe it accurately. In Figure 15, we can see that both models perform better in predicting harmony (left half of the

piano), but regarding melody (right half of the piano), the proposed model achieved higher F1-scores in general.

5.2.5 VPT state of the art

From Table 3 it may be inferred that the proposed model outperformed all other tested VPT methods on the PianoYT dataset. With out-of-distribution data such as PianoVal (Table 4), the transformer-based model achieved a higher precision than the proposed model. However, the proposed model achieves more accurate melodic transcription (described in Section 5.2.4) and overall a higher F1-score, which makes it generally more favorable. Thus, it can be stated and concluded, under the presented evaluation methodology and research assumptions, that the proposed model is the current state of the art for VPT.

5.2.6 Comparison to audio piano transcription

The authors of this work also compared audio-based transcription to the tested VPT methods on the PianoVal dataset in Table 4. Notice that this cannot be done for the PianoYT test set due to the fact that the labels are already generated by an audio-based transcription model (described in Section 3.1.1). It can be seen that currently, VPT is still inferior to audio-based transcription methods. Audio-based transcription achieved about 10% higher precision, recall and F1-score across the board. VPT still has a number of relevant use cases, however, such as the ones mentioned in Section 1.

5.2.7 Future research directions

The accuracy of VPT can be improved in multiple different ways. First, a dataset should be used that provides high-quality annotations, instead of pseudo-ground truth generated by audio-based transcription. Alternatively, the method described in Section 4.5.3 could be explored further as well.

To increase the temporal accuracy of the results, higher refresh-rate video should be used for training and inference. At a video frame rate of 30, some actions of the pianist can be missed if they are done in less than about 33 milliseconds. In contrast, inference in the audio-based method described in Section 3.4 used an audio sampling rate of 16000 Hz, which corresponds to a theoretical temporal accuracy of 0.0625 milliseconds. Neuromorphic cameras (event cameras) can be used to achieve such temporal accuracy for VPT too. These cameras detect ternary (negative, unchanged or positive) changes in brightnesses of pixels on the nanosecond-scale [48], which is fitting for the task of VPT, as it currently relies on pixel-level differences over time.

Additionally, the superior results of audio-based transcription discussed in Section 5.2.6 suggest that a hybrid model, adding audio input capabilities into the proposed approach, could be an area worth exploring. Finally, the models could be trained for longer, with more data and with additional modifications to their architectures.

6. Conclusion

In this work, four approaches for visual piano transcription were explored. Three related works were implemented for comparison and a new neural network architecture for visual piano transcription was proposed. All methods were trained using the PianoYT training dataset, which consists of approximately 20 hours of professional piano performances along with the corresponding note annotations.

The models were evaluated in two different scenarios. The first was the PianoYT test set, which is similar to the training dataset. To further assess the generalizability of the approaches, a new dataset, PianoVal, was created and used for evaluation.

The results indicate that the proposed model outperforms the current state of the art published by Zivanovic et al. [13], achieving an F1-score approximately 3 percentage points higher on the PianoYT test set and 7 percentage points higher on the PianoVal dataset, while also reaching nearly six times faster inference speed.

This work demonstrates that visual piano transcription is a promising step towards accurate piano transcription, especially in case of missing, noisy, or cluttered audio. When available, audio-based piano transcription remains superior in isolated environments, achieving about 10 percentage points higher F1-score compared to the proposed method. Future works should focus on increasing the temporal accuracy of VPT, creating higher quality video datasets, and exploring the fusion of audio-visual information. By advancing these approaches, it may become possible to transcribe any recorded piano performance accurately, improving the digitalization of acoustic piano pieces for educational and preservation purposes.

References

[1] He K., Zhang X., Ren S., and Sun J. Deep Residual Learning for Image Recognition. 2015. arXiv: 1512.03385 [cs.CV]. https://arxiv.org/abs/1512.03385.

[2] Hinton G., Deng L., Yu D., Dahl G. E., Mohamed A.-r., Jaitly N., Senior A., Vanhoucke V., Nguyen P., Sainath T. N., et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine* 29.6 (2012), pp. 82–97.

[3] Devlin J., Chang M.-W., Lee K., and Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 2019, pp. 4171–4186.

[4] LeCun Y., Bengio Y., and Hinton G. Deep learning. *nature* 521.7553 (2015), pp. 436–444.

[5] Deng J., Dong W., Socher R., Li L.-J., Li K., and Fei-Fei L. Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[6] Benetos E., Dixon S., Giannoulis D., Kirchhoff H., and Klapuri A. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems* 41 (2013), pp. 407–434.

[7] Benetos E., Dixon S., Duan Z., and Ewert S. Automatic music transcription: An overview. *IEEE Signal Processing Magazine* 36.1 (2018), pp. 20–30.

[8] Hawthorne C., Elsen E., Song J., Roberts A., Simon I., Raffel C., Engel J., Oore S., and Eck D. Onsets and Frames: Dual-Objective Piano Transcription. 2018. arXiv: 1710.11153 [cs.SD]. https://arxiv.org/abs/1710.11153.

[9] Emiya V., Badeau R., Daniel A., and David B. Automatic transcription of piano music. PhD thesis. Jan. 2008.

[10] Akbari M. and Cheng H. Real-time piano music transcription based on computer vision. *IEEE Transactions on Multimedia* 17.12 (2015), pp. 2113–2121.

[11] Saputra F., Namyu U. G., Vincent, Suhartono D., and Gema A. P. Automatic Piano Sheet Music Transcription with Machine Learning. *Journal of Computer Science* 17.3 (Mar. 2021), pp. 178–187. DOI: 10.3844/jcssp.2021.178.187. https://thescipub.com/abstract/jcssp.2021.178.187.

[12] Koepke A., Wiles O., Moses Y., and Zisserman A. Sight to sound: An end-to-end approach for visual piano transcription. *International Conference on Acoustics, Speech, and Signal Processing*. 2020.

[13] Zivanovic U. and Cancino-Chacón C. E. A Transformer-Based Visual Piano Transcription Algorithm. *arXiv preprint arXiv:2411.09037* (2024).

[14] Bay M., Ehmann A. F., and Downie J. S. Evaluation of multiple-f0 estimation and tracking systems. *ISMIR*. 2009, pp. 315–320.

[15] BT I. et al. Relative timing of sound and vision for broadcasting. *Relative timing of sound and vision for broadcasting* (1998).

[16] Olson D. L. and Delen D. Advanced data mining techniques. Springer Science & Business Media, 2008.

[17] Wu W., Huo L., Yang G., Liu X., and Li H. Research into the Application of ResNet in Soil: A Review. *Agriculture* 15.6 (2025). DOI: 10.3390/agriculture15060661. https://www.mdpi.com/2077-0472/15/6/661.

[18] Zhang J. Music genre classification with ResNet and Bi-GRU using visual spectrograms. *arXiv preprint arXiv:2307.10773* (2023).

[19] Chang P.-C., Chen Y.-S., and Lee C.-H. MS-SincResnet: Joint learning of 1D and 2D kernels using multi-scale SincNet and ResNet for music genre classification. *Proceedings of the 2021 international conference on multimedia retrieval*. 2021, pp. 29–36.

[20] Rezende E., Ruppert G., Carvalho T., Ramos F., and De Geus P. Malicious software classification using transfer learning of resnet-50 deep neural network. *2017 16th IEEE international conference on machine learning and applications (ICMLA)*. IEEE. 2017, pp. 1011–1014.

[21] Hochreiter S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02 (1998), pp. 107–116.

[22] Su K., Liu X., and Shlizerman E. Audeo: Audio generation for a silent performance video. *Advances in Neural Information Processing Systems* 33 (2020), pp. 3325–3337.

[23] Wang B. and Yang Y.-H. PerformanceNet: Score-to-audio music generation with multi-band convolutional residual network. *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 1174–1181.

[24] Tong Z., Song Y., Wang J., and Wang L. VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training. *arXiv preprint arXiv:2203.12602* (2022).

[25] Jocher G., Chaurasia A., and Qiu J. Ultralytics YOLOv8. Version 8.0.0. 2023. https://github.com/ultralytics/ultralytics.

[26] Kong Q., Li B., Song X., Wan Y., and Wang Y. High-resolution Piano Transcription with Pedals by Regressing Onsets and Offsets Times. *CoRR* abs/2010.01815 (2020). arXiv: 2010.01815. https://arxiv.org/abs/2010.01815.

[27] Edwards D., Dixon S., Benetos E., Maezawa A., and Kusaka Y. A Data-Driven Analysis of Robust Automatic Piano Transcription. Version 1.0.0. Feb. 2024. DOI: 10.5281/zenodo.10610212. https://doi.org/10.5281/zenodo.10610212.

[28] Hawthorne C., Stasyuk A., Roberts A., Simon I., Huang C.-Z. A., Dieleman S., Elsen E., Engel J., and Eck D. Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. *International Conference on Learning Representations*. 2019. https://openreview.net/forum?id=r1lYRjC9F7.

[29] Emiya V., Bertin N., David B., and Badeau R. MAPS-A piano database for multipitch estimation and automatic transcription of music (2010).

[30] Raffel C., McFee B., Humphrey E. J., Salamon J., Nieto O., Liang D., Ellis D. P., and Raffel C. C. MIR_EVAL: A Transparent Implementation of Common MIR Metrics. *ISMIR*. Vol. 10. 2014, p. 2014.

[31] Li Y., Yuan G., Wen Y., Hu J., Evangelidis G., Tulyakov S., Wang Y., and Ren J. EfficientFormer: Vision Transformers at MobileNet Speed. 2022. arXiv: 2206.01191 [cs.CV]. https://arxiv.org/abs/2206.01191.

[32] Steiner A., Kolesnikov A., Zhai X., Wightman R., Uszkoreit J., and Beyer L. How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers. 2022. arXiv: 2106.10270 [cs.CV]. https://arxiv.org/abs/2106.10270.

[33] Maurício J., Domingues I., and Bernardino J. Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: A Literature Review. *Applied Sciences* 13.9 (2023). DOI: 10.3390/app13095521. https://www.mdpi.com/2076-3417/13/9/5521.

[34] Lu L., Shin Y., Su Y., and Karniadakis G. E. Dying relu and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733* (2019).

[35] Xu B., Wang N., Chen T., and Li M. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853* (2015).

[36] Tomar S. Converting video formats with FFmpeg. *Linux Journal* 2006.146 (2006), p. 10.

[37] Buslaev A., Iglovikov V. I., Khvedchenya E., Parinov A., Druzhinin M., and Kalinin A. A. Albumentations: Fast and Flexible Image Augmentations. *Information* 11.2 (2020). DOI: 10.3390/info11020125. https://www.mdpi.com/2078-2489/11/2/125.

[38] Ravi N., Gabeur V., Hu Y.-T., Hu R., Ryali C., Ma T., Khedr H., Rädle R., Rolland C., Gustafson L., Mintun E., Pan J., Alwala K. V., Carion N., Wu C.-Y., Girshick R., Dollár P., and Feichtenhofer C. SAM 2: Segment Anything in Images and Videos. *arXiv preprint arXiv:2408.00714* (2024). https://arxiv.org/abs/2408.00714.

[39] Oquab M., Darcet T., Moutakanni T., Vo H. V., Szafraniec M., Khalidov V., Fernandez P., Haziza D., Massa F., El-Nouby A., Howes R., Huang P.-Y., Xu H., Sharma V., Li S.-W., Galuba W., Rabbat M., Assran M., Ballas N., Synnaeve G., Misra I., Jegou H., Mairal J., Labatut P., Joulin A., and Bojanowski P. DINOv2: Learning Robust Visual Features without Supervision. 2023.

[40] Lüddecke T. and Ecker A. Image Segmentation Using Text and Image Prompts. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 7086–7096.

[41] Ronneberger O., Fischer P., and Brox T. U-net: Convolutional networks for biomedical image segmentation. *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer. 2015, pp. 234–241.

[42] Takaki S., Nakashika T., Wang X., and Yamagishi J. STFT spectral loss for training a neural speech waveform model. *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 7065–7069.

[43] Engel J., Hantrakul L. (, Gu C., and Roberts A. DDSP: Differentiable Digital Signal Processing. *International Conference on Learning Representations*. 2020. https://openreview.net/forum?id=B1x1ma4tDr.

[44] Hawkins D. M. The problem of overfitting. *Journal of chemical information and computer sciences* 44.1 (2004), pp. 1–12.

[45] Alabdulmohsin I. M., Zhai X., Kolesnikov A., and Beyer L. Getting vit in shape: Scaling laws for compute-optimal model design. *Advances in Neural Information Processing Systems* 36 (2023), pp. 16406–16425.

[46] Carlson R. Melody Vs. Harmony: Similarities and Differences. https://www.hoffmanacademy.com/blog/melody-vs-harmony-similarities-and-differences.

[47] Miranda C. What is Melody? Discovering the Voice of Music. 2024. https://moises.ai/blog/tips/what-is-melody-in-music/.

[48] Rebecq H., Ranftl R., Koltun V., and Scaramuzza D. High speed and high dynamic range video with an event camera. *IEEE transactions on pattern analysis and machine intelligence* 43.6 (2019), pp. 1964–1980.

Appendices

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Karl Raud**,

*(*author's name)

1. grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the digital archives of the University of Tartu until the expiry of the term of copyright, my thesis

   Visual Piano Transcription,

   *(*title of thesis)

   supervised by Victor Henrique Cabral Pinheiro.

   *(*supervisor's name)

2. grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright;

3. am aware of the fact that the author retains the rights specified in points 1 and 2;

4. confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Karl Raud

15/05/2025