

Annotation Web - An open-source web-based annotation tool for ultrasound images

Erik Smistad^{*†‡}, Andreas Østvik^{*†‡}, Lasse Løvstakken^{*,†}

**Centre for Innovative Ultrasound Solutions (CIUS)*

Trondheim, Norway

†Norwegian University of Science and Technology (NTNU)

Dept. of Circulation and Medical Imaging

Trondheim, Norway

‡SINTEF Medical Technology

Trondheim, Norway

Abstract—The use of deep learning and other machine learning techniques requires large amounts of annotated image data. There exist several tools to annotate images, however to our knowledge there are no tools made specifically for medical ultrasound (US) images. As US is a temporal imaging modality it is essential for such a tool to handle image sequences/videos. To make annotation of US images as easy and efficient as possible, we made a web-based tool called Annotation Web (AW) enabling image classification, segmentation, bounding box and landmark annotation of ultrasound videos. We have now open-sourced this tool, making it available to everyone for free, and we invite the community to use it¹.

I. INTRODUCTION

Annotation is an essential part of supervised machine-learning. Annotation of images can be complicated and time-consuming and thus requires user-friendly and efficient tools.

In medical imaging, annotators are often clinicians who may not have high technical skills nor access to computers at the hospital where they can install software and store data. By making the system web-based, the users don't have to install anything, nor transfer any data. Everything is stored on the server and continuously saving the annotations allows busy clinicians to leave when they have to, and resume the annotation at a later time.

Ultrasound is a temporal imaging modality, and it is essential for annotators to see the ultrasound video playing, and being able to scroll back and forth, to know where the important structures/tissue types are in order to properly annotate the images. This is also an important feature for annotators to manually track speckle patterns which is often used to track tissue.

Annotation of medical images can be done in many different forms; image classification, bounding boxes, segmentation, landmarks etc. Each of these annotation tasks will have some

This research was funded by the Research Council of Norway under project 237887.

© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

¹<https://github.com/smistad/annotationweb/>

functionality that will be specific to that task only, and there will be some functionality that is common for all tasks.

There exist several free tools for annotating images and videos, such as the VGG Image Annotator (VIA) tool [1], and more specialized tools for annotating medical images such as 3D Slicer [2], ITK-Snap [3] and RIL-Contour [4]. However, to our knowledge there are no tools made specifically for medical ultrasound (US) images.

In this paper, we present a web-based tool for annotation of ultrasound image sequences called Annotation Web (AW). The system is available as open-source on GitHub with the permissive MIT license¹.

II. METHODS

To remove the burden of having to install software and transfer data, AW was made as a web-based system, so that the users only need a regular web browser to access and use it.

We will start by describing the backend running on the server side and the database structure. Then the frontend is described with focus on the fast ultrasound image playback and interactive annotation in the browser on the client side. Finally, some security aspects are discussed.

A. Backend - Django

Since many researchers today are using Python, we choose to use the Python web framework Django for the backend server-part of AW. Django is a high-level web framework focused on modular and fast application development. It also provides a secure user authentication system, with a flexible admin interface. Everything (e.g. image data, annotations, user information) is stored on the server side, and dynamically organized in a database. AW can either be run and used locally on a single machine, or deployed on a server and be accessible by multiple users over a network.

In order to create and organize datasets in AW, the actual data, the image sequences, must be on the machine running AW. Image sequences can be imported into AW as a series of images stored in metaimage (.mhd/.raw) format or as regular images (.png), or from videos. When importing from videos,

frames are extracted from the video and stored as series of images in the metainage format. This is done to speed up the data transfer during annotation. After the annotation is done, the actual annotations can be exported to disk. The format in which the annotations are exported in varies on the type of annotation. It is possible to create custom exporters, and to query the annotation data directly from the database.

For each annotation *task*, a set of *key frames* can be selected for each image sequence either by an administrator or by the annotator as part of the annotation task. Key frames, are frames which are supposed to be annotated as part of a task. An annotation task is assigned to a user, or multiple users for collaborative annotation. The annotator users will only see the tasks assigned to them.

In AW, we have tried to keep the task specific functionality and database tables separated from the common ones. A Django project consists of a set of apps, which should be as loosely coupled as possible, ideally it should be possible to add and remove apps with almost no code change. Thus in AW, every annotation task (classification, bounding box, segmentation, landmark..) are separate Django apps. This allows you to easily remove annotation apps, and to create new or copy an existing annotation app and modify it to your needs. We have used AW in several projects, often adapting the annotation apps to specific tasks.

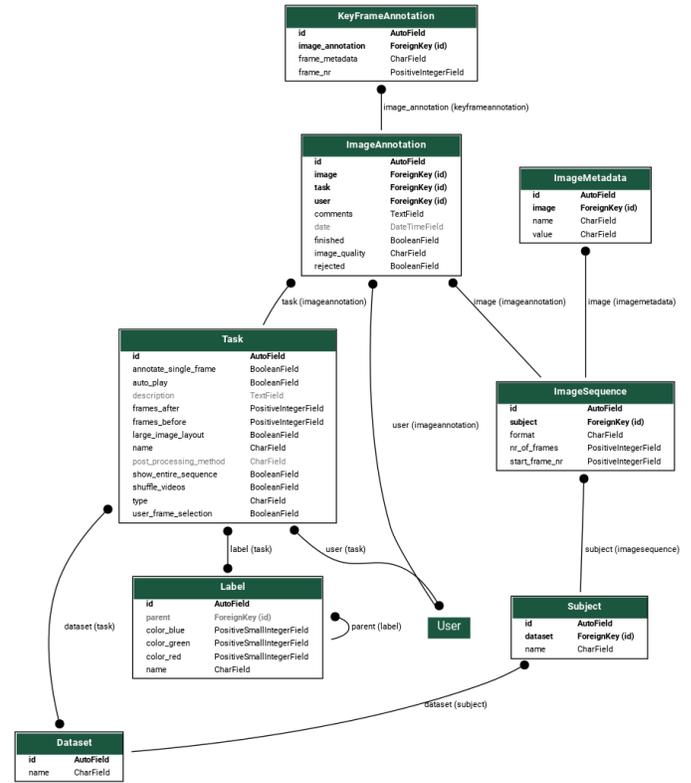


Fig. 1. Entity relation diagram of the core database of AW.

B. Database

Django uses a relational database with an object-relational mapping layer (ORM). By default AW uses a sqlite3 SQL database for storing all annotations and users. The images themselves are not stored in the database, but stored on disk with the path to the images stored in the database. Sqlite3 is a single file-based database, so the database is stored entirely in a single file, making it very easy to backup the database by copying this file. Still, AW and Django can be configured with several other SQL databases such as PostgreSQL and MySQL.

The ultrasound image data is organized in a hierarchy with a *Dataset*, consisting of several *Subjects* which can have multiple *Image Sequences*. An annotation *Task* is associated with one or more *Datasets*, *Users*, and *Labels*. To facilitate annotation of multiple key frames in a sequence, two entities representing annotations are used: *ImageAnnotation* which represents the annotation of an *Image Sequence* for a given *Task*, and *KeyFrameAnnotation* which represents the annotation of a specific frame. An *ImageAnnotation* can thus have multiple *KeyFrameAnnotations*. The entity relation (ER) diagram in Fig. 1 depicts the database structure of the AW core. In addition, each annotation task type (bounding box, landmark, segmentation and classification) have their own database objects storing task specific data which are connected to *KeyFrameAnnotation* and *Label*. Fig. 2 shows an ER diagram of these task specific entities.

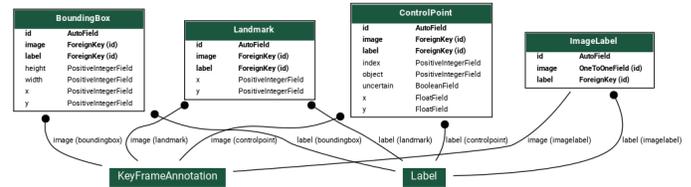


Fig. 2. Entity relation diagram of the annotation task specific database of AW. Annotation data for the bounding box, landmark, segmentation and classification tasks are connected to the *KeyFrameAnnotation* and *Label* objects of the AW core.

C. Frontend - Ultrasound playback and interactive annotation in the browser

To enable real-time video playback and at the same time be able to annotate/draw on top of the video, we used HTML canvas to draw the US images and annotations in the browser using Javascript. Another option could have been to use the HTML video element to have playback of the ultrasound video. However, this has several drawbacks: 1) visualizing annotations on top of the video would be difficult, 2) videos often compress the data, 3) there exist many video codecs and proper playback requires the user to have the right codec installed, 4) ultrasound recordings are not always stored as videos, and would therefore need to be converted to a video. Thus, we instead decided to use HTML canvas to draw the ultrasound images in the browser using Javascript. This may

sound complicated, but is actually simple. With HTML canvas, we can draw arbitrary 2D geometry on top of the ultrasound image using plain Javascript: bounding boxes, splines, lines, dots etc.

When a user starts to annotate an ultrasound sequence, the browser will load every image in the sequence using HTTP requests, and will store the ultrasound images in memory in the client’s browser. Thus, when the ultrasound sequence is finally loaded, playback is very fast and smooth since all the data is stored in the client’s memory. This also enables annotators to move quickly between frames using the mousewheel and arrow keys for efficient manual speckle tracking. However, if you have very long ultrasound sequences, memory usage and loading time can become an issue. This might be solved in the future by loading/streaming parts of long image sequences on demand instead. The images are currently transferred after PNG lossless compression has been applied. Compressing the images with lossy JPEG before transfer could be used to speed up the data transfer, but this has not been implemented yet, as image quality was considered more important.

D. Security

When dealing with medical data, security is always important. Note that any system connected to a network will never be 100% secure, and AW comes with absolutely no guarantees. We use Django’s built-in user authentication system together with the two-factor authentication plugin `django-otp` to restrict access to authorized users only. We recommend only storing anonymized or deidentified data on the system, and when deploying the system, use a firewall/virtual private network, as well as HTTPS/SSL encryption. More information on this can be found on GitHub.

III. RESULTS

AW has been used successfully in our research groups since 2016 in several ultrasound machine-learning projects such as segmentation of the left ventricle [5], cardiac view classification [6] and detection of nerves and blood vessels [7].

Figures 3 to 6 show screenshots of four different annotation tasks currently implemented in AW. On the left the user can select key frames, save or reject the annotation, set the image quality, select label etc. The right side shows the ultrasound image with the annotations drawn on top. The cardiac segmentation task is an extension of the spline segmentation, providing specialized features such as a simple M-mode line to help identify ED/ES frames, and ensuring splines of the different structures are connected in a standardized manner.

IV. DISCUSSION

A simple web-based annotation tool for medical ultrasound images was presented. Compared to the VGG Image Annotator [1] which also supports video annotation, AW is server-based thereby removing the burden of data storage and transfer from the annotators which we believe is useful in a medical setting with large datasets. However, having an annotation

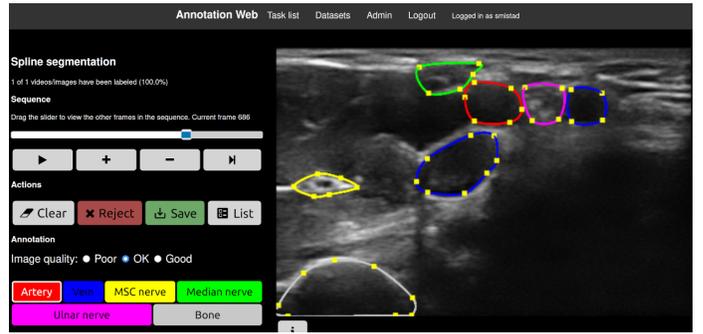


Fig. 3. Segmentation using splines.

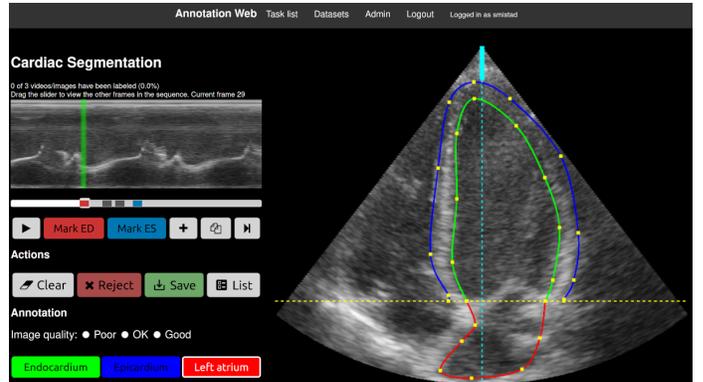


Fig. 4. Cardiac segmentation using splines with a simple M-mode line through the image center.

system on an online server with medical data adds several security and privacy concerns which must be addressed by its users. Compared to medical annotation software such as 3D Slicer, ITK-Snap and RIL-Contour [2]–[4], AW is focused on simplicity and being web-based removes the burden of installation and data storage and transfer. We also believe that by using Python and Django, compared to C++, AW is simpler to customize and extend.

Future work includes adding more annotation types, better support for long image sequences, and other useful features such as semi-automatic annotation using trained machine-learning models, improved dataset search and filtering, and image post processing.

V. CONCLUSION

We have presented, Annotation Web, an easy-to-use annotation tool for US images which is open-source, fast and scalable. It is available on GitHub, together with documentation for getting started, using and deploying the system. The tool has several annotation tasks implemented such as image classification, segmentation using splines, cardiac left ventricle segmentation, landmark and bounding box, supporting a wide range of US image analysis problems. The system was made to be highly modular making it easy to customize and develop new and specialized annotation tasks.

