Scoliosis Subteam Final Presentation

Eric Chen, Nathan Zhong, Lucas Yim, Austin Peng, Brian Zhang, Sai Nikil Vangala, Patrick Weng, Charlotte Hettrich, Daniel You, Minseung Jung, Pranav Malireddy, Aalekhya Narayani

Overview

- Introduction
- Datasets
- EMADE Preparation
- Midterm Status
- EMADE progress
- Primitive Progress
    - Image Processing
    - Models
- Future Work

Introduction

Introduction

Adolescent Idiopathic Scoliosis (AIS) - Most common form of scoliosis affecting children ages 10-18

Cobb Angle - The maximum angle formed by the two most tilted vertebrae in the spine

- Angles of at least 10 are considered scoliosis
- Between 20 and 40 requires back bracing
- Greater than 40 necessitates surgery

Motivation - Measurements between doctors can and do vary; even measurements taken by the same doctor throughout the day can change.

Our goal: Use EMADE to explore the search space of existing machine learning models for cobb angle prediction and image processing primitive to improve performance.

Datasets

AASCE

- Set of 580 x-ray images
- Each image contains:
  - 68 landmark points (4 for each vertebra)
  - 3-region Cobb angles
- Evaluated over symmetric mean absolute percentage error (SMAPE) for cobb angle
- Many models used MSE for landmark loss

Shriners

- Set of 1000 x-ray images
- Only extracted Cobb angles from 300 image instances
- No landmark data
- Performed cropping to standardize data for models
- Will be evaluated over SMAPE and MSE (implemented in EMADE)

Data Packing

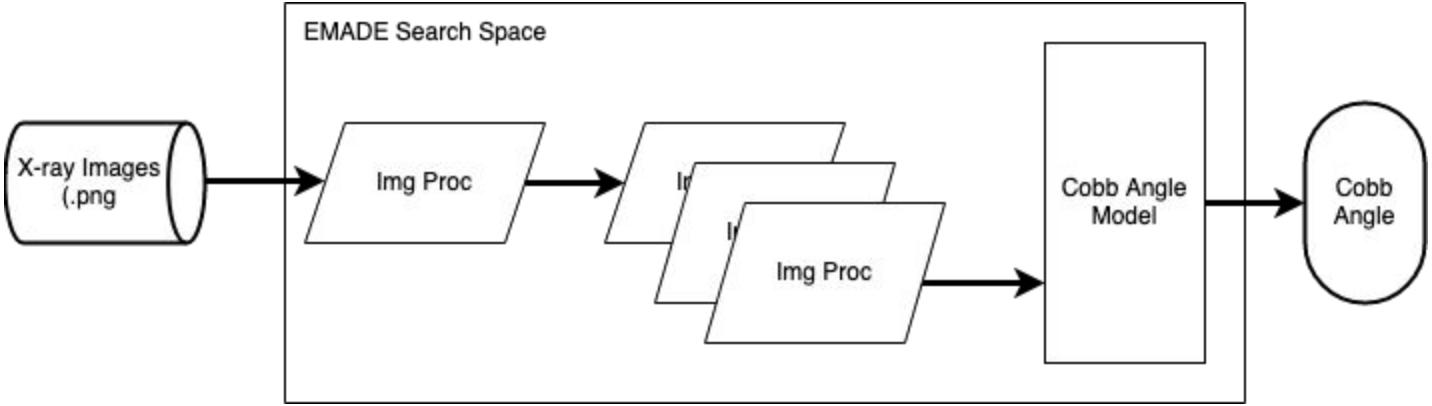
- Both AASCE and Shriners datasets were packed into .npz format to be utilized with EMADE
  - kept the original AASCE train/test/validation splits
  - created a 70/15/15 split for the Shriners dataset
- Each training instance consists of the image data (np.array) and the maximum Cobb angle (int)

Midterm Status

Where we were:

- In the process of setting up EMADE on Azure
- Collected a set of image processing and cobb angle model candidates
- Received additional ~700 images from Shriners; no easy way to extract truth data
- Pinned down the EMADE pipeline

The Pipeline



EMADE

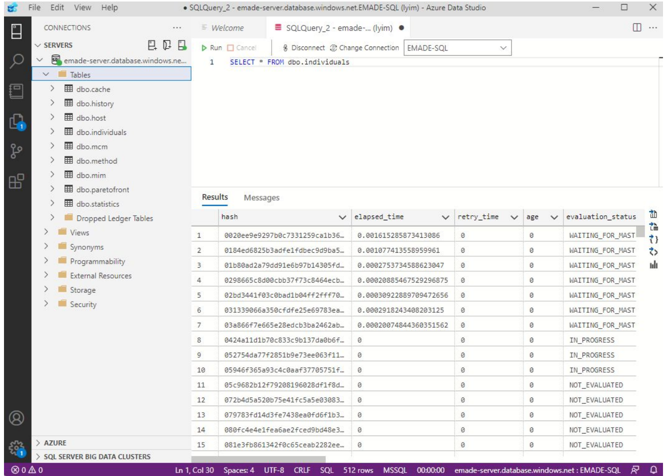
Azure SQL Setup

AzureSQL (MSSQL)

- EMADE typically runs on MySQL databases
  - Needed to change the source code to allow EMADE to work with MSSQL
- Tried Pyodbc and PyMssql with SQLalchemy
  - Ran into driver compatibility errors with Pyodbc
- Changes will be pushed onto scoliosis-CacheV2 after some cleanup

AzureSQL DB setup

- Create new usernames and passwords for the database
- Give each user read and write permissions for the database
- Give the users ALTER and CONTROL permissions for each schema you need
    - (default is dbo)
- View database through Azure Data Studio



MySQL to MsSQL Mapping

- Configuration changes in sql_connection_orm_base.py
- Mapped the pickle type to an IMAGE type to comply with MsSQL compared to the original MEDIUMBLOB type

Source Type	MySQL Type	Comment
IMAGE	TINYBLOB/MEDIUMBLOB/LONGBLOB	Depending on its length

Source: https://dev.mysql.com/doc/workbench/en/wb-migration-database-mssql-typemapping.html

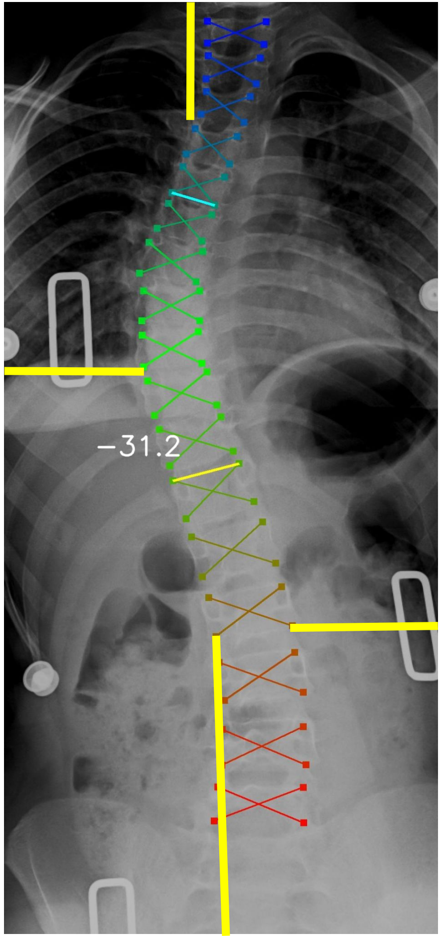
Image Processing Primitives

Cropping

- Initial Cropping Formula Formula:

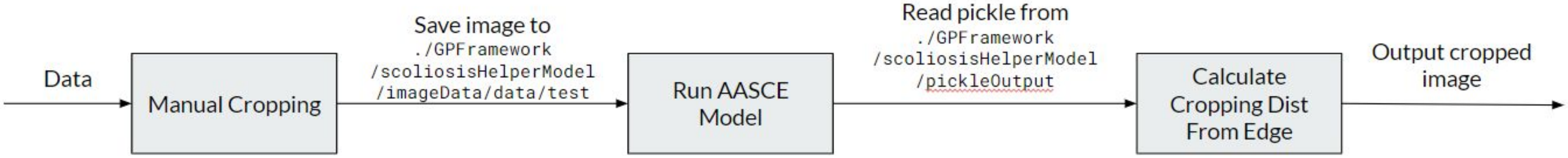
  Sides: (img.width - 1000) / 2 - 100
  Top: (img.height - 2300) / 3 - 300
  Bottom: (img.height - 2300) * 2 / 3 - 200

- Found the average distance between the edge of the image and the top/bottom/left/right most points for the AASCE images excluding the top three and bottom three vertebrae
- For each Shriner's image, found the top/bottom/left/right most points from the initial cropping predictions, found the distance between those points and the edge of the ORIGINAL picture, and cropped the image so that the distances match the averages from AASCE



Cropping Primitive

- New file: `src/GPFramework/cropping_methods.py`
- Follows the same cropping algorithm as the slide before
- Used `RegistryWrapperS` as my wrapper and has primitive take in a numpy array of ints and output a numpy array
- Utilizes AASCE model which is housed in `src/GPFramework/scoliosisHelperModel`
- Note: Only runs on Python 3.6 and 3.7; Python 3.8 will cause torch error



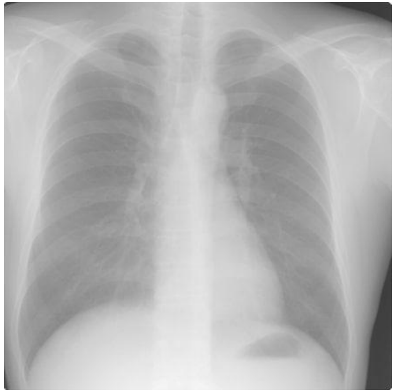
Unit Testing Cropping Primitive

- New unit tests written in `./GPFramework/UnitTests/cropping_methods_unit_test.py`
- First unit test checks that primitive correctly crops image (`test_spine_xray_cropping_single_image`)
    - Results in Azure
- Second unit test checks primitive handles npz packaged data (`test_spine_xray_cropping_multiple_image`)
    - Unit test took almost 5 minutes to run, processing just 18-20 images
    - May be due to the fact we have to save every image file and read the pickle output; will be a direct area of improvement we will look into next semester
- Future Concerns:
    - Making the process more efficient and streamlined
    - Generalizing it to all images and not just specifically Shriner's

Edge Sharpening

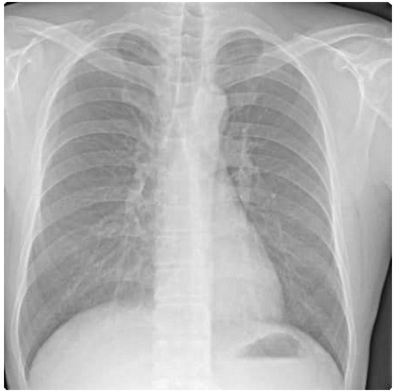
- As the Shriner's images had much less defined spine and vertebrae when compared to AASCE, edge sharpening techniques were explored to see if it improves.
- Three image enhancement algorithms were explored:
  - Contrast Limited Adaptive Histogram Equalization (CLAHE)
    - Histogram-based method used to improve contrast in images
  - Unsharp Masking (UM)
    - Linear filter that is capable of amplify high-frequencies of an image.
  - High-Frequency Emphasis Filtering (HEF)
    - Gaussian High Pass Filter to emphasize and accentuate the edges.

Edge Sharpening Examples



Original

CLAHE



UM

HEF

Image from the National Sciente Foundation Multimedia Gallery

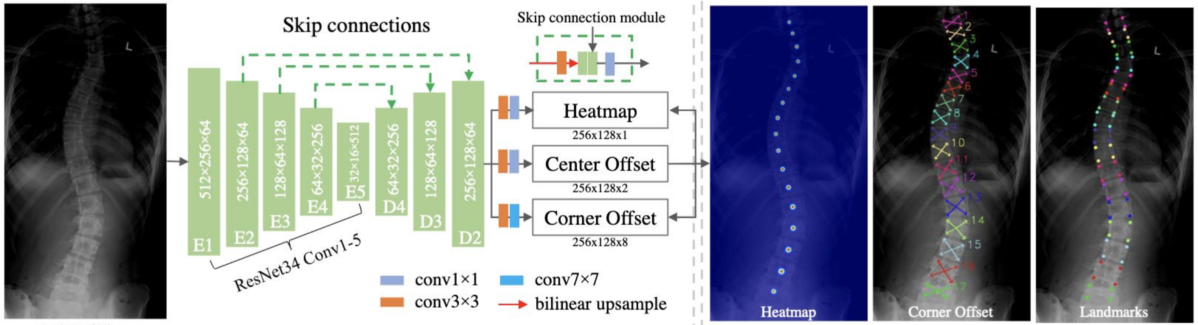
Edge Sharpening(Primitives)

- All primitives use `RegistryWrapperS as a wrapper, and take/return a numpy array of ints`
- `Each primitive is in their own file under src/GPFramework and share utility methods found in src/GPFramework/edge_sharpening_utils.py`
- CLAHE
  - The algorithm takes in three parameters: window_size, clip_limit, and n_iter
    - Set to 40, 4, and 1, respectively based on recommendations from the creators of these algorithms
- UM
  - The algorithm takes in three parameters: filtering, radius, and amount
    - Set to 1, 5, and 2, respectively based on recommendations from the creators of these algorithms
- HEF
  - The algorithm takes in one parameter: d0v
    - Set to 45 based on recommendations from the creators of these algorithms

Model Primitives

Vertebra-Focused Landmark Detection Model

- Proposed a method predicting cobb angles based on identifying centerpoints of vertebrae, then finding corresponding corner points, and finally calculating cobb angle.
- Challenge winner: 21.375
- Paper Model: 10.31



1024×512 (a) Vertebra-Focused Landmark Detection Network (b) Landmark Decoding

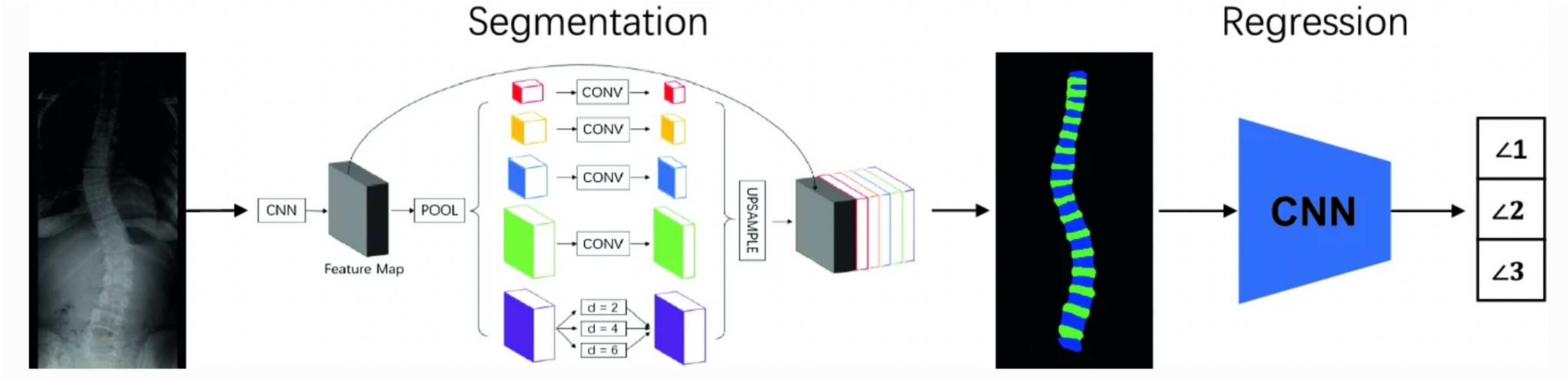
VFL Model Primitive Progress

- The model and pretrained weights successfully added as a primitive into EMADE
  - Input: numpy array from EmadeDataPair(changed to jpg)
  - Output: Cobb angle
  - Unit tests outputted a Cobb angle of about 58.5 degrees on our test image
- Wrapped with RegistryWrapperS and primitive registry updated with the model primitive

```
loaded weights from /home/azureuser/cloudfiles/code/Users/emade-git/
processing 0/1 image ...
processing image: scoliosis.jpg
predicted angle:  58.54664750282438
Results 58.54664750282438
```

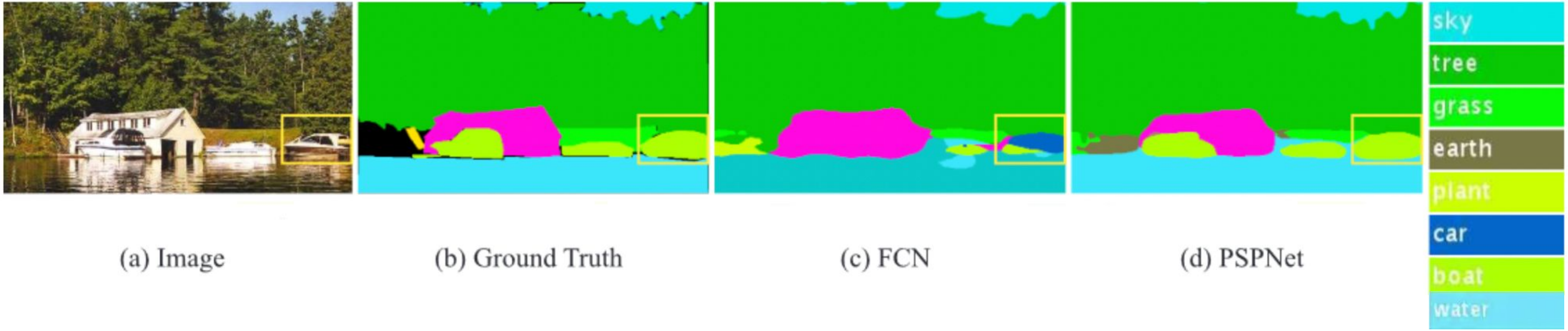
Seg4Reg

- Performs vertebrae segmentation followed by Cobb Angle regression
- Preprocessed images with random rescale, rotation, and Gaussian noise
- Used a neural network to generate segmentation masks on the vertebrae
- Used another neural network to predict Cobb Angles off of segmentation masks



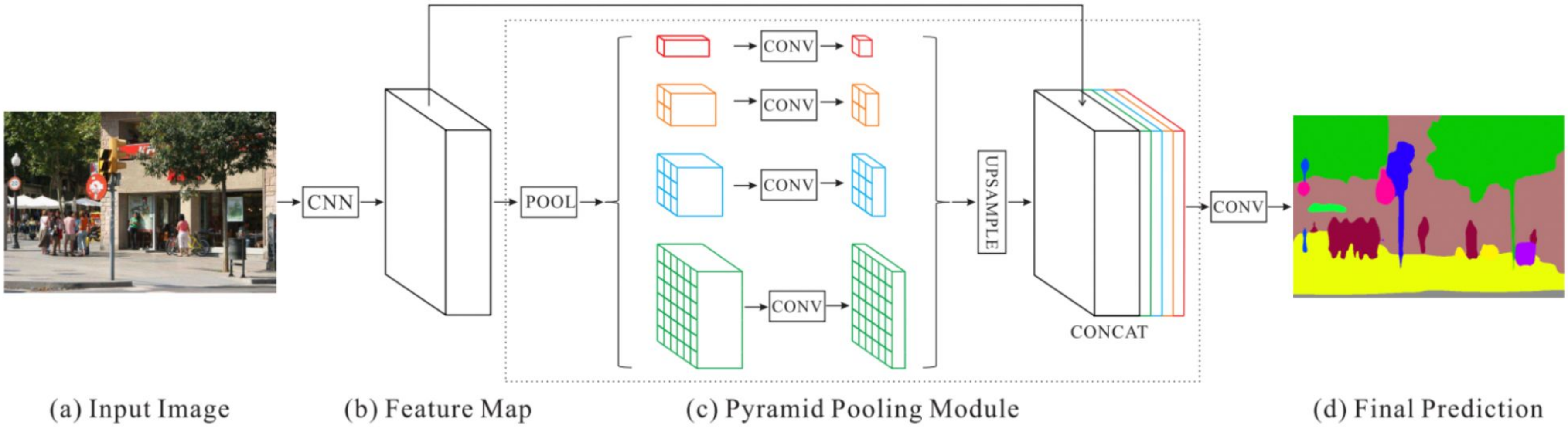
PSPNet (Image Segmentation)

- Backbone architecture is ResNet-101 (skip layers to overcome the vanishing gradient problem)
- Uses a pyramid pooling scheme to exploit global context information using different region-based aggregation



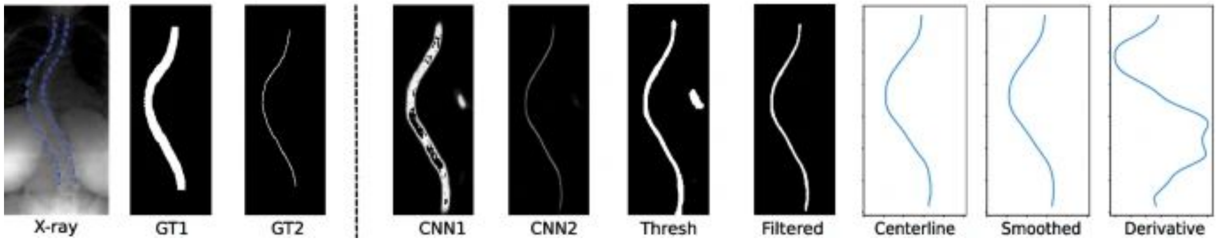
(a) Image (b) Ground Truth (c) FCN (d) PSPNet

PSPNet (Image Segmentation)



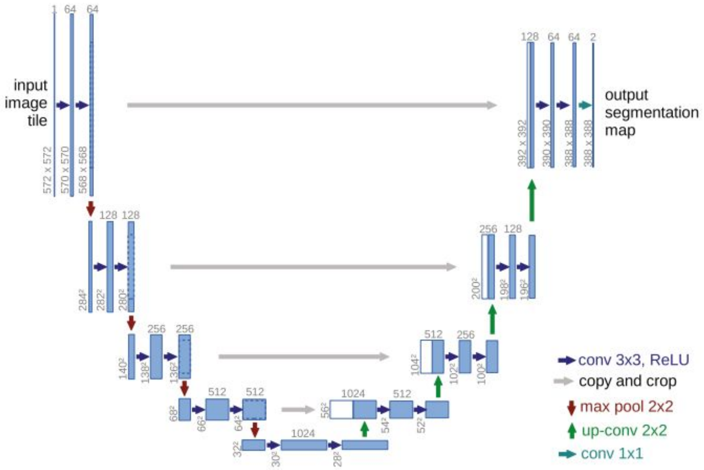
Centerline Extraction Cascaded NNs

- Cascaded Neural Nets derive a centerline from the image.
- Postprocessing steps guarantee continuity of the curve, detecting the sides of the centerline, and finally selecting points between the sides as the final centerline curve.
- The cascaded nature of the neural nets allowed the model to better handle contrast
- Running into difficulties with image size; paper mentions manual cropping for evaluation



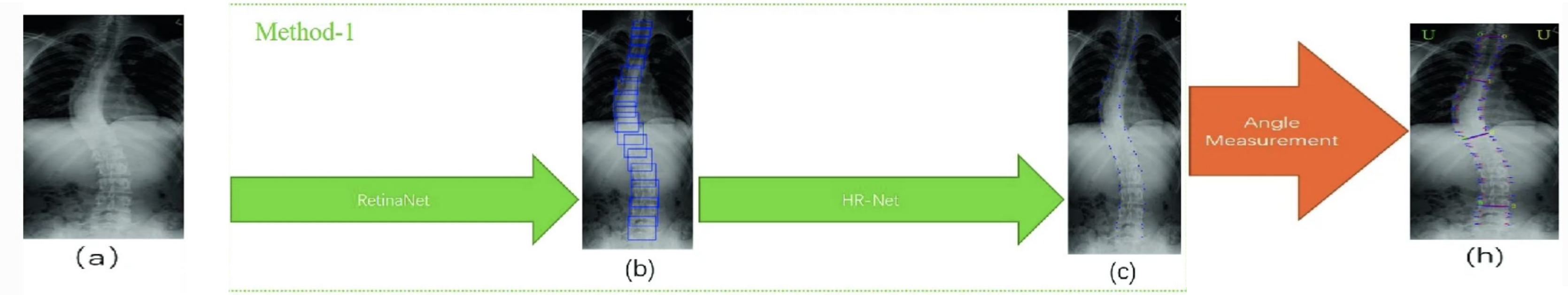
U-Net

- **Overview:** popular, state-of-the-art deep convolutional network for image segmentation
  - Combines a contracting network with successive upsampling layers, creating the U-Shape
  - Propagates context information to higher layers to increase resolution of features
- **Progress:**
  - Original U-Net as developed by the original authors replicated using the Keras API running on Tensorflow
    - Crop and copy operation replaced with concatenation layers
  - Successful on the ISBI dataset for cell segmentation using generated weights and also trained with data augmentation
  - Future Work:
    - Develop bounding boxes code and/or vertebrae localization to specifically adapt to scoliosis problem



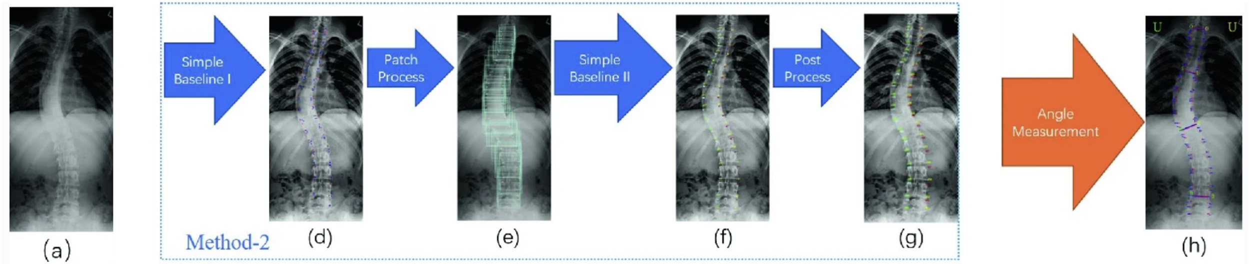
Accurate Automated Keypoint Detection

- Method 1
  - RetinaNet & HR-NET
  - Bounding boxes (68 keypoints) using RetinaNet
  - Detect the 4 key points of the bounding box using HR-NET
  - ResNet-50-FPN backbone: Trained for 180k iterations with a total of 8 images per minibatch



Accurate Automated Keypoint Detection

- Method 2
  - Simple Baseline I: to grasp the global implicit sequentiality of key points
  - Patch Process: detect key points by Simple Baseline I
  - Simple Baseline II: to make highly robust model
  - Post Process: clusters and removes outliers, 2 results obtained
  - ResNet-152 backbone
- Fusion Method



Future Work

Preprocessing

Image Cropping:

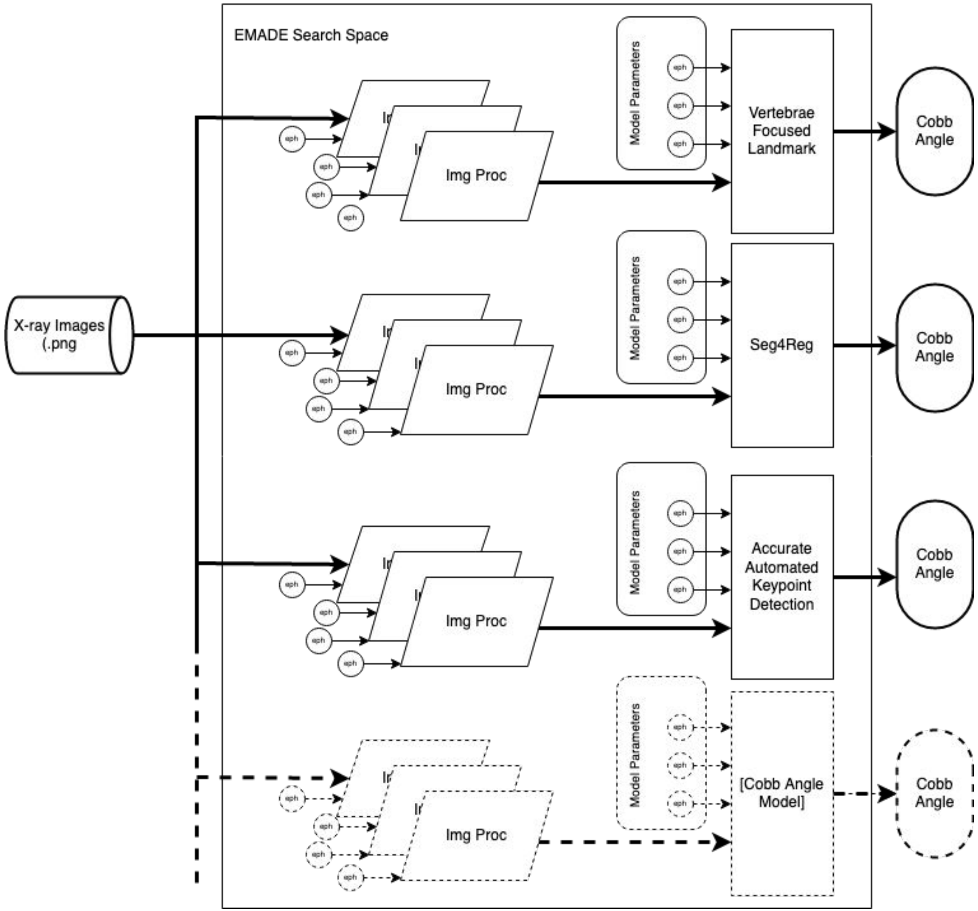
- Improve image cropping efficiency by altering AASCE model to take in numpy array and return landmarks directly
- Generalizing image cropping primitive to apply to all x-ray images rather than just Shriner's

Edge Cropping:

- Ensuring that the correct edge cropping mechanism is chosen for a set of x-ray images.
- Generalizing edge sharpening primitive to apply to all x-ray images rather than just Shriner's

Models

- Implement other models and parameterize their inputs for EMADE to optimize over and expand search space



Others

- Customize multiple evaluation functions for each model for training
- Many models are based on neural nets; potentially try applying a similar approach to Image Processing team's old neural network based search space
- There are additional metrics we would like to measure in the future
- Integrate the 700 images into our Shriners dataset
- Expand the image processing set
- Categorize cobb angle ranges to turn problem into classification
- Add dynamic preprocessing functionality to EMADE to randomize training preprocessing

References

Papers

- Yi, Wu, Huang, Qu, Metaxas, “Vertebrae-Focused Landmark Detection For Scoliosis Assessment”
- Ronneberger, Fischer, Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”
- Horng, Kuok, Fu, Lin, Sun, “Cobb Angle Measurement of Spine from X-Ray images using Convolutional Neural Network”
- VFL Repo: https://github.com/yijingru/Vertebra-Landmark-Detection
- Chen, K., Peng, C., Li, Y., Cheng, D., Wei, S. (2020). Accurate Automated Keypoint Detections for Spinal Curvature Estimation. In: Cai, Y., Wang, L., Audette, M., Zheng, G., Li, S. (eds) Computational Methods and Clinical Applications for Spine Imaging. CSI 2019. Lecture Notes in Computer Science(), vol 11963. Springer, Cham. https://doi.org/10.1007/978-3-030-39752-4_6
- Ronneberger, Fischer, Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”
- U-Net Implementation: GitHub - zhixuhao/unet: unet for image segmentation