**\* What are the major goals of the project?**

As shown in Figure 001 (attached), there are three major goals of this project:

* Phase 1: Hardware Acquisition
* Phase II: Data Development
* Phase III: Algorithm Development

This first phase consists of hardware procurement, development and installation. This required managing a complex multi- organizational relationship between the vendor (Aperio/Leica Biosystems), the Temple Hospital Information Technology group (TUHS-IT) and Temple University Main Campus Information Technology (TUMC-IT).

Subgoals for this phase of the project included:

* Procurement of Scanner Installation and Verification User Training
* Procurement of Network Storage Installation and Verification
* Final Hardware Certification

The second phase of the project consisted of data development. This involved working closely with the vendor to certify and integrate the slide scanner hardware and software. Subgoals for this phase of the project included:

* Preliminary Archival Scanning User Acceptance Testing Production Archival Scanning Workflow Integration
* User Acceptance Testing Production Scanning IRB Application
* Preliminary Database Release User Acceptance Testing Production Database Release

The third phase of the project consisted of algorithm development. This involved the development of a deep learning-based system to automatically classify data. Subgoals for this phase of the project included:

* Pilot Experiments System Tuning
* System Performance Analysis Physician Feedback
* Final System Performance Evaluation Physician Acceptance Testing

**\* What was accomplished under these goals and objectives (you must provide information for at least one of the 4 categories below)?**

**Major Activities:**

The first two goals of the project collectively result in our ability to collect data. Over the course of the project we have digitized over 100,00 slides. We have not released all of this data yet because we are still in the process of renaming and organizing it. However, as described in various sections of this report, we have released two important subsets.

The average image size is 372 Mbytes, which confirms the fact that these are extremely high-resolution images. The total size of our released data is close to 7 Terabytes. We have reconfigured our web server to support the dissemination of these large datasets. While this activity is a detail beyond the scope of the report, it was an important step to allow people to efficiently download such a large amount of data.

We have also trained undergraduate students to annotate breast tissue data. We have believe we have this process under control and are generating high quality annotations. We have since expanded into the next largest subset of the data - urinary prostate slides. To do this, we had to learn how to classify these slides. We created a collaboration with the Pathology Department at Fox Chase Cancer Center (FCCC), a center of excellence for cancer treatment in the U.S.

The third task, the development of software to automatically classify slides, has resulted in a release of an open source system that we refer to as a Digital Pathology Cancer Recognition System (DPATH-CRS). An artificial neural network (ANN) is used to classify the regions of the digital pathology slides. An ANN tries to assimilate the recognition ability of a pathologist. ANNs have many free parameters that need to be tuned to the specific applications - a process is called training. Then this trained ANN is evaluated by classifying some unseen data. If it can guess the correct classes, then it can be employed as a diagnostic system, and if not, the training process should be repeated.

We designed a five-step ANN training and evaluation pipeline:

1. PATCH EXTRACTION

A digital pathology slide file contains several images with different resolutions but from a single specimen. Annotators, that are trained to diagnose the regions, attach the labels to some selected regions, but not all parts of a slide. Pathologists usually look at the very subtle details of the slide to diagnose the region. Therefore, the processing is done on the highest resolution which is called level 0. Level 0 contains a very large image, such as 50Kx50K pixels.

Patches are square-shaped images that are cropped from some parts of level 0 and processed as an atom. This means that a patch is assumed to contain a single class, not different classes, and the diagnosis system assigns a single label to it. Annotated areas are made with freehand drawing tools, so they can have any shape. Also, all patches in an annotated area have a single label. A rolling window is used and the window size is selected to be 256x256, with a step size of 256 in both directions. Every window that has more than 50% overlap with the annotated area is assumed to be an acceptable patch and is stored as a 256x256 TIFF image in the corresponding patches directory.

At this point, all annotated areas from all slides are broken down into patches, and patches are stored in the corresponding directories.

2. PATCH DISTRIBUTION

Patch distribution is necessary to allocate different slides to the training, development, and evaluation steps of an ANN. Two considerations make the intelligent distribution of the patches crucial. First, the ratio of patches in every class in comparison to the total patches must be kept approximately similar. It means that if the ratio of patches of class 1 in the training dataset is 10%, then it is expected to have the same percentage in the development and evaluation dataset. Second, to have an open-loop (blind) development and evaluation, there must be different slides in different datasets and no similar slides should be in two different datasets.

To satisfy these two constraints, a greedy algorithm is used. Since solving this problem in an optimal way is an NP-hard problem, to make the whole process fast, we have to relax some constraints. The second constraint is crucial, so we relaxed the first one. Instead of maintaining all classes' ratios fixed, just the classes with the smallest percentages are tried to be kept fixed. Hence, a greedy search algorithm tries to maintain the percentage of classes in order. It distributes the slides that contain the rarest class with predefined ratios. Then, it distributes the slides with the second rare classes, again with predefined ratios, but with priority given to training, development, and evaluation, respectively.

3. TRAINING

At this point, the slides that are used for training are known, but the number of patches in the classes varies, resulting in an unbalanced dataset. Two approaches to solve this problem were examined: weighted loss and random sampling. The weighted loss function assumes that the prior probabilities of classes in the datasets are proportional to the number of patches in that class. This proportional weighting can be used in either original format or after a softmax operation.

The second approach is based on random sampling. The idea comes from the observation that most of the patches in the outnumbered classes are very similar, especially in the patterns. So, it is rational that instead of using all the available patches in these outnumbered classes, a limited number of them are selected randomly and then train the ANN on this random subset of that class, while keeping all the rare classes as they were. This approach is called random sampling.

Combining the random subset approach with the weighted loss function solves the highly unbalanced dataset issue. But how effective is random sampling. To find out, the ANN is trained on some random subset of the training dataset, then the one that has the best loss on the development set is selected as the best trained ANN. Since the random subset of the training dataset is usually much smaller, the training finishes fast, but the development process takes more time because the ANN is examined on all the patches on the development dataset.

An ANN model needs to be predetermined before the training. The convolutional layers in the image analysis with the deep learning methods are proven to be very effective for making a pyramidal representation of an image. Therefore, most image analysis systems use some convolutional filtering in a hierarchical order or in a parallel format for extracting the patterns in different resolutions and making a new set of features. Then with the help of one or two fully connected layers, they classify these features. We follow the same approach, but use a pre-trained ANN. This approach is known as transfer learning. We use a very well-known, simple, and very effective kind of these network which is called Residual Network or ResNet.

4. DECODING

Given an unannotated whole slide, it is cropped into too many patches. These patches are shown to the neural network and the ANN classifies every patch, which is called decoding. In the decoding phase, some minor errors can be pruned. We can define a minimum size for every region. For example, a cancerous region in the size of a single patch should not be surrounded by all the normal neighbors and it is expected that some similar patches make a large area. So, a 2D moving average filter or a 2D Gaussian filter with a predefined radius is used to estimate the center patch and this estimation is being compared with the decoded label. If they are close to each other, nothing is changed, but if they are too different, the estimated output of the filter is used instead of the decoded decision.

5. EVALUATION

Evaluation needs annotated data. So, only the decoded parts that contain annotations are used. A confusion matrix is generated that is very essential evaluating the effectiveness of this system.

**ggggggg**

**\* What was accomplished under these goals and objectives (you must provide information for at least one of the 4 categories below)?**

**Major Activities:**

The first two goals of the project collectively result in our ability to collect data. Over the course of the project we have digitized over 100,00 slides. We have not released all of this data yet because we are still in the process of renaming and organizing it. However, as described in various sections of this report, we have released two important subsets.

The data can be characterized as follows:

1. Breast Tissue (6%)

2. Urinary Prostate (36%)

3. Gastrointestinal (18%)

4. Lymph Nodes (8%)

5. Other (32%)

The average image size is 372 Mbytes, which confirms the fact that these are extremely high-resolution images. The total size of our released data is close to 7 Terabytes. We have reconfigured our web server to support the dissemination of these large datasets. While this activity is a detail beyond the scope of the report, it was an important step to allow people to efficiently download such a large amount of data.

We have also trained undergraduate students to annotate breast tissue data. We have believe we have this process under control and are generating high quality annotations. We have since expanded into the next largest subset of the data - urinary prostate slides. To do this, we had to learn how to classify these slides. We created a collaboration with the Pathology Department at Fox Chase Cancer Center (FCCC), a center of excellence for cancer treatment in the U.S. Dr. Yulan Gong met with our undergraduates and trained them to annotate these slides.

The third task, the development of software to automatically classify slides, has resulted in a release of an open source system that we refer to as a Digital Pathology Cancer Recognition System (DPATH-CRS). An artificial neural network (ANN) is used to classify the regions of the digital pathology slides. An ANN tries to assimilate the recognition ability of a pathologist. ANNs are a very non-linear mathematical function that maps the input data (such as an image) to their classes. This function has too many free parameters that need to be tuned to the specific applications; this tuning process is called training. Then this trained ANN is evaluated by asking to make a decision on some unseen data. If it can guess the correct classes, then it can be employed as a diagnostic system, and if not, the training process should be repeated.

We designed an ANN training and evaluation pipeline to make a cancer diagnosis system to assimilate the pathologists' skills. There are five steps in this pipeline:

1. PATCH EXTRACTION

A digital pathology slide file contains several images with different resolutions but from a single specimen. Annotators, that are trained to diagnose the regions, attach the labels to some selected regions, but not all parts of a slide. So, the slides are not totally annotated, but partially annotated.

Pathologists usually look at the very subtle details of the slide to diagnose the region. Therefore, the processing is done on the highest resolution which is called level 0. Level 0 contains a very large image, such as 50Kx50K pixels. The OpenSlide library is used to read the content of a slide. Every pixel has four channels, RGBA. But the alpha channel is not usually being used. So, three channels, RGB, are enough to keep the data for every pixel.

Patches are square-shaped images that are cropped from some parts of level 0 and processed as an atom. This means that a patch is assumed to contain a single class, not different classes, and the diagnosis system assigns a single label to it. Therefore, the size (width and height) of these elemental patches are highly dependent on the application. On the other hand, the training needs annotated data and the slides are partially annotated, hence, the patches are extracted from the regions that are annotated and the un-annotated parts are not used for further processing.

For every slide in the dataset, a different directory is made with the same name as the slide file. For every possible class, a subdirectory is made in the slide directory. Patches with similar labels from a slide are stored in the corresponding label subdirectory in the corresponding slide directory. If some labels do not exist in a slide, then that subdirectory remains empty.

Annotated areas are made with freehand drawing tools, so they can have any shapes. Also, all patches in an annotated area have a single label. A surrounding rectangle of the area cropped out of level 0. A rolling window is swept from the upper left point of this surrounding rectangle to the upper right and then one step down and repeat until all the surrounding rectangle is swept. Based on the observations, the window size is selected to be 256 in 256 and the step size is 256 in both directions. So, the patches will not have overlap. Every window that has more than 50% overlap with the annotated area is assumed to be an acceptable patch and is stored as a 256 in 256 TIFF image in the corresponding patches directory.

Finding if a window is in the annotated area or not is very time-consuming, so the slide is rescaled to (1/256, 1/256) and a binary mask is made with a logical value of 1 as annotated and 0 as unannotated. Then a window with size 256 in 256 can be assumed as a single logical value, 1 means annotated and 0 means unannotated. Scaling a large image might be too time-consuming, but the surrounding rectangles are usually very smaller than the whole slide. Moreover, the scaling is done with the NearestNeighbor method which is computationally very lightweight.

At this point, all annotated areas from all slides are broken down into patches, and patches are stored in the corresponding directories.

1. Patch Distribution

 Patch distribution is necessary to allocate different slides to the training, development, and evaluation steps of an ANN. Two considerations make the intelligent distribution of the patches crucial. First, the ratio of patches in every class in comparison to the total patches must be kept approximately similar. It means that if the ratio of patches of class 1 in the training dataset is 10%, then it is expected to have the same percentage in the development and evaluation dataset. Second, to have an open-loop (blind) development and evaluation, there must be different slides in different datasets and no similar slides should be in two different datasets.

To satisfy these two constraints, it needs an NP-hard search space. To make the whole process fast, we have to relax some constraints. The second constraint is crucial, so we relaxed the first one. Instead of maintaining all classes' ratios fixed, just the classes with the smallest percentages are tried to be kept fixed. Hence, a greedy search algorithm is developed. It tries to maintain the percentage of classes in order. It distributes the slides that contain the rarest class with predefined ratios. Then, it distributes the slides with the second rare classes, again with predefined ratios, but with priority given to training, development, and evaluation, respectively.

Therefore, a greedy algorithm with two given priorities, e.g. maintaining the rarer class percentage and the order of the training, development, and evaluation datasets distributes the slides into three datasets in a fast semi-optimal way.

1. Training

 At this point, the slides that are used for training are known. But the number of patches in the classes is usually very different, sometimes on the scale of 1 to 10000. This situation is usually called the training in an unbalanced dataset.

Training operation is optimizing a loss function with the steepest gradient descent algorithm or one of its successors, such as Adam. Optimization of the loss function in a highly unbalanced dataset reaches severe bias toward the classes with an exceeded number of samples and it never tries to learn the patterns of classes with a few samples.

To solve the issue with the unbalanced dataset two approaches are examined; weighted loss and random sampling. The weighted loss function assumes that the prior probabilities of classes in the datasets are proportional to the number of patches in that class. This proportional weighting can be used in either original format or after a softmax operation. Either way, the ANN is forced to consider more loss for classes with less number of patches. Then some impacts of the unbalanced number of patches alleviate, but not completely because the datasets are highly unbalanced.

The second approach tries to solve this highly unbalanced impact. The idea comes from the observation that most of the patches in the outnumbered classes are very similar, especially in the patterns. So, it is rational that instead of using all the available patches in these outnumbered classes, a limited number of them are selected randomly and then train the ANN on this random subset of that class, while keeping all the rare classes as they were. This approach is called random sampling.

Combining the random subset approach with the weighted loss function solves the highly unbalanced dataset issue. But how effective a random sampling is. To find out, the ANN is trained on some random subset of the training dataset, then the one that has the best loss on the development set is selected as the best trained ANN. Since the random subset of the training dataset is usually much smaller, the training finishes fast, but the development process takes more time because the ANN is examined on all the patches on the development dataset.

The model of an ANN needs to be predetermined before the training. The convolutional layers in the image analysis with the deep learning methods are proven to be very effective for making a pyramidal representation of an image. Therefore, most image analysis systems use some convolutional filtering in a hierarchical order or in a parallel format for extracting the patterns in different resolutions and making a new set of features. Then with the help of one or two fully connected layers, they classify these features. We follow the same approach, but instead of designing and training an ANN from scratch, a pre-designed and pre-trained ANN is being used. This approach is usually known as transfer learning. It means that the ANN which is being trained on a different but large set of images retrained again on a specific set of images to be tuned on a new domain.

ImageNet is a large competition and the dataset is fully populated with natural images and too many classes. Several large ANNs are trained on this large dataset and are now openly available to be fine-tuned on different specialized domains. We use a very well-known, simple, and very effective kind of these network which is called Residual Network or ResNet. The residual name comes from the fact that the output of every layer of the neural network consists of its processed output and its raw input which is called residual. These residuals make the pyramidal multiresolution representation of the input image and cause the neural network to converge faster because they prevent vanishing gradients.

 A pretrained ResNet18 on the ImageNet is trained on our training dataset. For every epoch of the training, blind evaluation is employed on a random subset of the development dataset. The best model is selected based on the least weighted loss on the random development sub-dataset. Several of these ResNet18s are trained on different random sub-datasets of the training dataset, and the one with the least weighted loss on the development dataset is kept as the best model.

1. Decoding

Given an unannotated whole slide, it is cropped into too many patches. These patches are shown to the neural network and the ANN classifies every patch, which is called decoding. The output of decoding is a mask of size (1/256, 1/256) of the original level0 of the slide with categorical (or indexed) pixels. This mask is usually small and can be seen as a whole in every simple image viewer.

In the decoding phase, some minor errors can be pruned. We can define a minimum size for every region. For example, a cancerous region in the size of a single patch should not be surrounded by all the normal neighbors and it is expected that some similar patches make a large area. So, a 2D moving average filter or a 2D Gaussian filter with a predefined radius is used to estimate the center patch and this estimation is being compared with the decoded label. If they are close to each other, nothing is changed, but if they are too different, the estimated output of the filter is used instead of the decoded decision.

So, it depends on how much contrast is accepted; the pure output of decoding or the smoothed decision. If the decoding output seems to be too noisy, then smoothing is necessary.

1. Evaluation

Evaluation needs annotated data. So, only the decoded parts that contain annotations are used. The decoded mask and annotated area are compared with each other. If they match, the detection is done perfectly, but if they differ, it is important to know the actual class is confused with which other class. This makes the confusion matrix. In a confusion matrix, actual labels are located in the rows, and detected labels are located in the columns. Every element (i, j) in the confusion matrix shows how much class i is confused by class j.

The confusion matrix is very essential to find out the effectiveness of many clinical computer-aided diagnosis (CAD) systems. In most clinical applications, the pure accurate diagnosis is not the only parameter that defines the cost. There is not a perfect diagnosis, so in every diagnosis, the cost of correct and incorrect diagnosis must be considered. For example, incorrect diagnosing of a normal tissue as malignant may cause financial loss, but diagnosing a cancerous tissue as normal may cost a human life. The confusion matrix can be combined with some clinical considerations to further process the detections. This post-processing makes a scientific model to be applicable in real-world applications and can be our future investigations.

**Ggggggggggggggg**

**Clinical Impact:**

**Tumor Board stuff…**

**Gggggggg**

**Specific Objectives:**

**Specific Objectives:**

The specific objectives for the project were essentially: (1) procure and deploy research instrumentation that supported the digitization of pathology slides; (2) the development of a large corpus of annotated pathology slides that will support machine learning; and (3) develop deep learning software that automatically classifies and segments pathology slides, so that pathologists can integrate these classifications into their workflows, thereby improving their efficiency.

**Significant Results:**

Data Collection:

In the third year of this project, we originally planned to focus on software development and annotation of data, while we kept data collection running as a mature task.

Unfortunately, due to COVID-19, data collection was more or less suspended in March 2020 because it was no longer possible for our undergraduates to work at Temple University Hospital. At the time we were on track to reach 100,000 slides by the end of 2020.

We currently have digitized over 60,000 slides. These can be categorized as follows:

1. Breast Tumor (6%)

2. Urinary Prostate (36%)

3. Gastrointestinal (18%)

4. Lymph Nodes (8%)

5. Other (32%)

We have completed annotation of the Breast Tumor subset (3,640 slides) and will release these in early 2021. We are in the final stages of deidentifying the medical reports, which will be released with the data. This subset totals 1.23 Terabytes of data. The average image size is 372 Mbytes, which confirms the fact that these are extremely high-resolution images.

Part of our preparations to release this data included an extensive reconfiguration of our web server to support the dissemination of these large datasets. While this activity is a detail beyond the scope of the report, it was an important step to allow people to efficiently download such a large amount of data. We increased the amount of web space on the web server and reconfigured the anonymous rsync capability so thatmultiple downloads could be managed efficiently using advanced disk caching strategies.

We also began annotating the next largest subset of the data - urinary prostate slides. To do this, we had to learn how to classify these slides. We created a collaboration with the Pathology Department at Fox Chase Cancer Center (FCCC), a center of excellence for cancer treatment in the U.S. Dr. Yulan Gong met with our undergraduates and trained them to annotate these slides. This training was completed at the end of 2020. We are now in the early stages of annotating this data.

In the process of creating this collaboration, we reached an agreement with FCCC to digitize pathology slides in their tumor bank, collaborating with Dr. Denise Connolly. FCCC maintains one of the largest repositories of this type of data in the country. It is extremely well-curated. We spent 9 months working through IRB issues with them (this is always a very slow process), but should begin transferring slides in Spring 2021 and continue digitization throughout 2021. The agreement for this data sharing has been approved by Temple University and is awaiting approval by FCCC. This is an enormous opportunity since these slides are very well documented and will serve as important ground truth data for machine learning. It will also allow us to expand the types of stains seen in our database, which will make the machine learning problem more difficult and yet more clinically relevant.

We are in the process of collaborating with FCCC on some future research opportunities in the pathology area and are very excited about the opportunity to bring them into this research area. Their pathology expertise will be extremely valuable.

We are developing an extensive annotation guidelines document that includes examples of each type of annotation performed. The first version of this document will be released with the breast tumor subset. It will be publicly available and open for community review. A preliminary version of this document is available at this

URL: https://www.isip.piconepress.com/publications/reports/2021/tuh\_dpath/annotations/. However, this document is not yet ready to be widely disseminated. It is interesting to note this document was written by the undergraduate annotation team, which is an extremely competent group of students.

**Key outcomes or Other achievements:**

**The specific objectives for the third year of this project were:**

**1. Digitize 100,000 images by December 2020.**

**2. Annotate and release several subsets of data.**

**3. Release software to automatically classify images.**

**4. Acquire user feedback on the value of the software.**

**We have digitized 60,000 slides and were on track for 100,000 before COVID-19. We have requested, and received, an extension to the project to allow us to continue digitizing images.**

**We have also created a data pipeline with Fox Chase Cancer Center that should pay long-term dividends. Slide scanning should commence in Spring or Summer 2021 depending on COVID-19.**

**We are on the verge of releasing our first large subset of data - the Breast Tumor Corpus - which will contain 3,640 annotated images.**

**We have developed software that can classify breast tumor data with high accuracy. We are working to release this software in early 2021.**

**The breast tumor data is a substantial contribution because nothing like it exists in the open source. We expect it will quickly become quite popular.**

**The creation of a strategic partnership with FCCC should pay dividends in the long run as it gives us access to another data source.**

**This project has employed four undergraduates in three capacities: web developer, application programmer and software engineer. The student doing web development had the opportunity to learn state of the art methods in full stack web development including using the package Boost to develop and maintain web pages. The application engineer position had the opportunity to learn a broad range of skills including real-time DSP programming, streaming interfaces, Unix shell programming and, of course, implementation of signal processing and deep learning in Python. The software engineering position involved learning how to design extensible software systems than facilitate the integration and adaptation of research software. Two graduate students were employed on the project. They were tasked with the development of the software tools, which involved how to apply and optimize deep learning software in Python. All students in the project also learned how to organize and manage data sets. All students have also been trained on how to present their work at a professional conference, and how to document their work. We use software management systems and a weekly reporting structure that emphasize good technical communication.**

**\* Have the results been disseminated to communities of interest? If so, please provide details.**

All data, resources, and software are available from the project web site as described in the original proposal. The project web site is located at https://www.isip.piconepress.com/projects/nsf\_dpath/. The data is also crossreferenced on our generate data and resources site: https://www.isip.piconepress.com/projects/nedc/. We provide direct downloads and also support an anonymous rsync server (nedc@www.piconepress.com).

We also maintain a listserv we use to communicate with people. Anyone can sign up at this URL: https://isip.piconepress.com/projects/nsf\_dpath/html/request\_access.php