TRAVA (www.travasecurity.com)

potential CS advisors: M. Al Hasan, G. Mohler, M. Dundar, S. Mukhopadhyay, A. Durresi, Q. Hu, Y. Liang, Y. Xia

1. Overall scoring

Develop an approach and algorithm to generate an overall cyber risk score based on results from various assessment types, including: vulnerability assessments (measuring known vulnerabilities in an environment); cyber maturity questionnaires (surveys that classify how well an organization is adhering to industry standard security recommendations); and, phishing simulations (measuring how members of an organization respond to simulated phishing messages). While each of these assessments can be classified separately, a higher-level classification is needed to reflect overall maturity/posture.

2. Open source vulnerability scanner

Research available open-source tools for vulnerability scanning, compare those to commercially available scanners, and prototype their usage in a variety of environments. This would include external scanning, internal discovery, internal scanning as well as agent-based scanning. The goal is to identify vulnerability scanning solutions that leverage community input on the latest issues, and can be used in a cost-effective way.

3. Prioritization of remediation steps

Develop a method for prioritizing cyber security mitigation recommendations, in an automated fashion based on several factors. Those factors would include: current risk tolerance based on industry and reliance on critical data, regulatory or contractual requirements, current security posture as measured by vulnerability assessment tools, cyber maturity surveys and phishing simulations. As a result, a prioritized list of recommended steps could be presented to an organization to help them improve their overall cyber risk.

4. Processing and classification of insurance policies

Research a process to take copies of cyber insurance policies from clients, analyze them for types and level of coverages and compare them to industry standards. From this comparison, develop a rating system to classify cyber coverage, and to generate recommendations for a client to improve their insurance coverage.

5. Security visualization

Develop an approach to visualizing cyber security posture and risk, to help clients clearly understand where they currently stand, where gaps exist and what areas they can focus on to improve their security posture. Visualizations would be based on data that is collected/measured from customers, as well as industry standards and frameworks. The result would be a set of visualizations that could be presented to clients that would be intuitive and instructive, and would not require the client to have a deep understanding of security concepts to interpret.

ClearObject (www.clearobject.com)

Potential CS advisors: J. Zheng, M. Tuceryan, M. Dundar, G. Mohler, S. Mukhopadhyay

1. Pipe - Root condition clock position

Every root defect condition during a pipe inspection can cover a different area within the pipe and must be noted by an inspector using a clock position to represent the area of the damage type. However, with our AI model identifying conditions the inspector missed we will need to also be able to identify these clock positions. The objective of this work would be to be able to use our existing pipe roots model as an input, and output the answer in terms of a clock position.

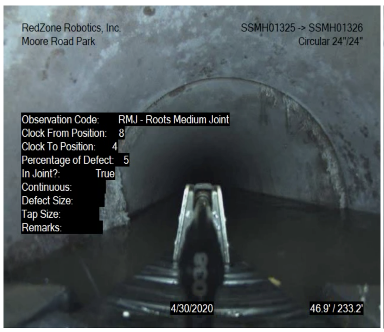


Image 1: In this example the inspector denotes the given damage (RMJ) using the clock position from 8 to 4.

Image 2: This is the camera in the same position as image 1 prior to the inspector identifying the condition. We would expect our root model to draw the bounding box around the defect, and then this project would try to identify those clock positions.

Image 3: This is an example of our roots model and its ability to identify the defect condition across frames with confidence and draw the bounding boxes.

2. Pipe – Size of root condition

This use case is similar to the root condition clock position, however now we are looking to identify the size of the roots represented as a % of the pipe blocked by the root. The objective of this work would be to be able to use our existing pipe roots model as an input, and add the ability classify the amount of blockage caused by the root.

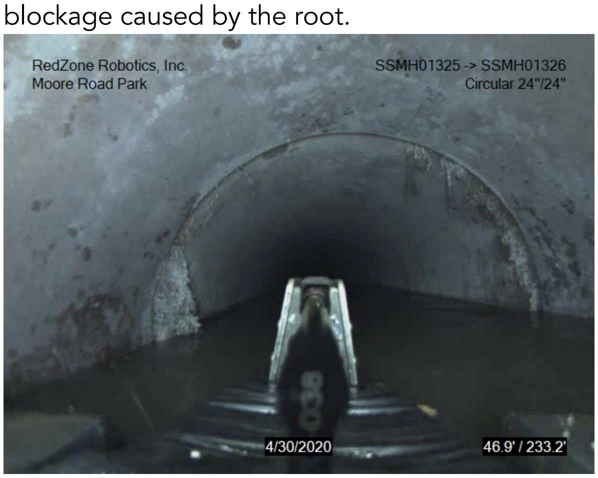


Image 1: In this example you see an example where there are multiple smaller roots spread across the joint so the % at this joint may be a combination of the multiple roots.

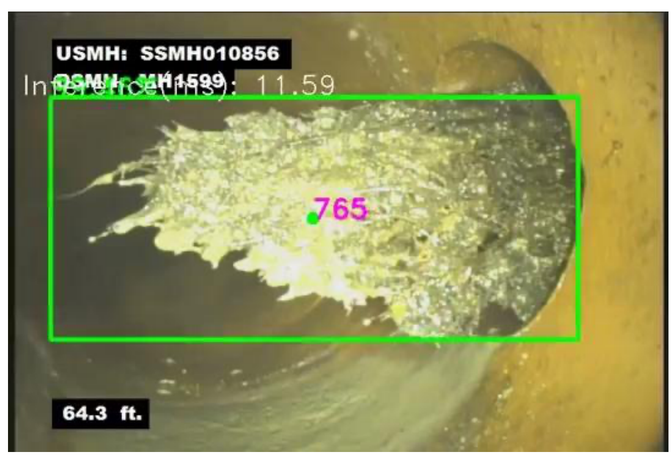


Image 2: This is an example of our roots model and its ability to identify the defect condition across frames with confidence and draw the bounding boxes. This would add a new measure to the bounding box giving us an expected % of the pipe blocked by the root.

3. Pipe – Tap identification

Develop an object detection model to identify taps during an inspection.

Image 1: In this example you see an example of what a tap looks likes from a distance in a DI (ductile iron) pipe.

Image 2: In this example you see an example of what a tap looks likes from a distance in a VCP (Vitrified Clay Pipe) inspection.

Image 3: In this example you see an example of what a tap looks likes as you begin to get closer on a VCP inspection/

4. Pipe – Joint identification

Develop a vision model that accurately identifies each joint in a given pipe. This will be used to help map out a digital representation of the entire inspection.

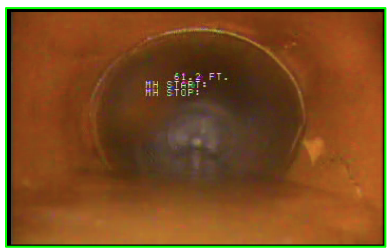


Image 1: In this example you see an example of what a tap looks likes from a distance. The other images here show examples of how a tap looks when the robot gets closer.

Image 2: This is an example of our roots model and its ability to identify the defect condition across frames with confidence and draw the bounding boxes. This would add a new measure to the bounding box giving us an expected % of the pipe blocked by the root.

Image 2: This is an example of our roots model and its ability to identify the defect condition across frames with confidence and draw the bounding boxes. This would add a new measure to the bounding box giving us an expected % of the pipe blocked by the root.

5. Pipe – OCR distance tracking

Develop an ocr model that tracks the distance covered by the inspection so that a distance can be tied back to each frame. This output will be used to create a very detailed map of the pipe inspection. Note: Each inspection software is a little different o n how the distance appears on the screen during inspection. We'll also want to account for frames where the OCR may ready incorrectly in order to reduce noise in output. Also, the output will not be linear as there are times where the camera will pause at given distances.

Image 1: This is an example where the distance of the video is shown in the top right corner, so the output of this frame would be 22.5'.

Image 2: This is an example where the distance of the video is shown in the bottom right corner, so the output of this frame would be 46.9'.

6. Pipe – OCR pipe size analysis

Develop an ocr model that identifies the pipe size of the inspection based on the embedded text shown at the beginning of every inspection video. This output will be used to create a very detailed map of the pipe inspection. Note: Each inspection software is a little different on how the pipe size appears on the screen during inspection setup.

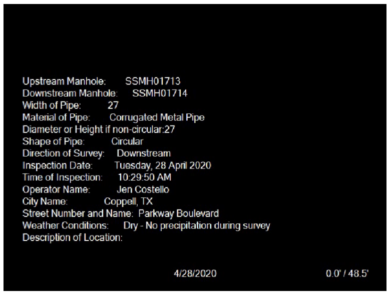


Image 1: This is an example where the size of the pipe in the video is shown in the 3rd line at the start of the video, so the output of this frame would be 27.

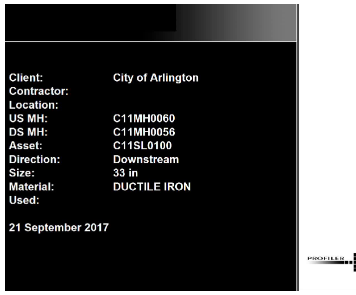


Image 2: This is an example where the size of the pipe in the video is shown in the 8th line at the start of the video, so the output of this frame would be 33.

Image 3: This is an example where the width of the pipe in the video is shown in the 11th line at the start of the video, so the output of this frame would be 6.

7. Pipe – OCR inspection information

Develop an ocr model that identifies the inspection details typically displayed at the early stages of the inspection video, and report output in a consistent matter, no matter the inspection software that was used. This output will be used to validate the data information received in the corresponding database files. Note: Each inspection software is a little different on how the pipe inspection information appears on the screen during inspection setup.

8. Pipe – Material type identification

Develop a classification model that identifies the given pipe material for each frame of an inspection video. Also detect when change of material occurs.

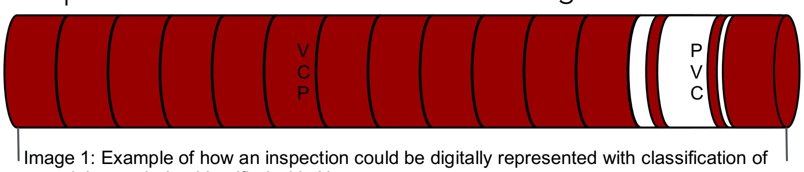


Image 1: Example of how an inspection could be digitally represented with classification of material types being identified with AI.

Image 2: Example of what VCP looks like

Image 3: Example of what DI looks like

Image 4: Example of what PVC looks like

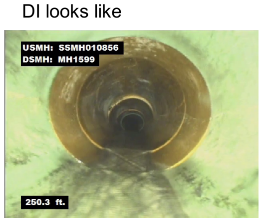


Image 5: Example of what a material change from PVC to VCP looks like.

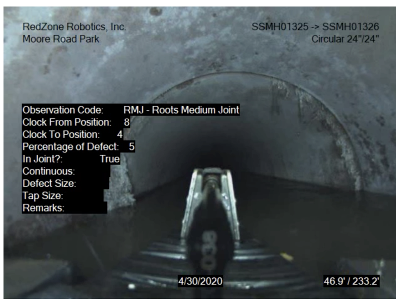
	PACP Material Field 34
CODE	**ENGLISH**
ABS	Acrylonitrile Butadiene Styrene
AC	Asbestos Cement
BR	Brick
CAS	Cast Iron
CLC	Clay-lined Concrete Pipe
CMP	Corrugated Metal Pipe
CP	Concrete Pipe (non-reinforced)
CSB	Concrete Segments (bolted)
CSU	Concrete Segments (unbolted)
CT	Clay Tile (not vitrified clay)
DIP	Ductile Iron Pipe
FRP	Fiberglass Reinforced Pipe
OB	Pitch Fiber (Orangeburg)
PCP	Polymer Concrete Pipe
PCCP	Pre-stressed Concrete Cylinder Pipe
PE	Polyethylene
PP	Polypropylene
PSC	Plastic/Steel Composite
PVC	Polyvinyl Chloride
RCP	Reinforced Concrete Pipe
RPM	Reinforced Plastic Pipe (Truss Pipe)
SB	Segmented Block
SP	Steel Pipe
VCP	Vitrified Clay Pipe
WD	Wood
XXX	Not Known
ZZZ	Other (State in Additional Information field 59)

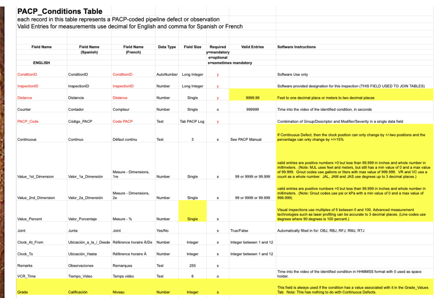
9. OCR material type vs actual pipe material type

Take the information from the OCR Inspection Information model and compare that with the Material Type Identification model and identify inspections where the two do not match. We have found instances where the inspector manually entered the wrong material type information during setup, so the inspection information does not accurately reflect the material being inspected. By identifying this issue we can take the necessary steps to resolve the issue. Resolutions could be an auto-correction of the inspection meta data, or having a user manually review and confirm the necessary changes.

10. OCR manual condition identification

Every inspection has a human performing it, and during that inspection they use the software to annotate when they identify conditions. The software captures this information and embeds it on the shared inspection videos. Therefore, our system can be used to automatically capture these manual conditions so we can use them to create a digital inspection record that in the future may no longer require the database file to also be shared. Note: Each inspection software is a little different on how the pipe inspection information appears on the screen so creating a consistent format for each condition is needed.





PACP_Conditions Table
each record in this table represents a PACP-coded pipeline defect or observation
Valid Entries for measurements use decimal for English and comma for Spanish or French

Field Name ENGLISH	Field Name (Spanish)	Field Name (French)	Data Type	Field Size	Required y=mandatory o=optional s=sometimes mandatory	Valid Entries	Software Instructions
ConditionID	ConditionID	ConditionID	AutoNumber	Long Integer	y		Software Use only
InspectionID	InspectionID	InspectionID	Number	Long Integer	y		Software provided designation for this inspection (THIS FIELD USED TO JOIN TABLES)
Distance	Distancia	Distance	Number	Single	y	9999.99	Feet to one decimal place or meters to two decimal places
Counter	Contador	Compteur	Number	Single	o	999999	Time into the video of the identified condition, in seconds
PACP_Code	Código_PACP	Code PACP	Text	Tab PACP Log	y		Combination of Group/Descriptor and Modifier/Severity in a single data field
Continuous	Continuo	Défaut continu	Text	3	s	See PACP Manual	If Continuous Defect, then the clock position can only change by +/-two positions and the percentage can only change by +/=15%
Value_1st_Dimension	Valor_1a_Dimensión	Mesure - Dimensions, 1re	Number	Single	s	99 or 9999 or 99.999	valid entries are positive numbers >0 but less than 99.999 in inches and whole number in millimeters. (Note: MJL uses feet and meters, but still has a min value of 0 and a max value of 99.999. Grout codes use gallons or liters with max value of 999.999. VR and VC use a count as a whole number. JAL, JAM and JAS use degrees up to 3 decimal places.)
Value_2nd_Dimension	Valor_2a_Dimensión	Mesure - Dimensions, 2e	Number	Single	s	99 or 9999 or 99.999	valid entries are positive numbers >0 but less than 99.999 in inches and whole number in millimeters. (Note: Grout codes use psi or kPa with a min value of 0 and a max value of 999.999)
Value_Percent	Valor_Porcentaje	Mesure - %	Number	Single	s		Visual inspections use multiples of 5 between 0 and 100. Advanced measurement technologies such as laser profiling can be accurate to 3 decimal places. (Line codes use degrees where 90 degrees is 100 percent.)
Joint	Junta	Joint	Yes/No		s	True/False	Automatically filled in for: OBJ, RBJ, RFJ, RMJ, RTJ
Clock_At_From	Ubicación_a_la_/_Desde	Référence horaire À/De	Number	Integer	s	Integer between 1 and 12	
Clock_To	Ubicación_Hasta	Référence horaire À	Number	Integer	s	Integer between 1 and 12	
Remarks	Observaciones	Remarques	Text	255	s		
VCR_Time	Tiempo_Video	Temps vidéo	Text	6	o		Time into the video of the identified condition in HHMMSS format with 0 used as space holder.
Grade	Calificación	Niveau	Number	Integer	s		This field is always used if the condition has a value associated with it in the Grade_Values Tab. Note: This has nothing to do with Continuous Defects.

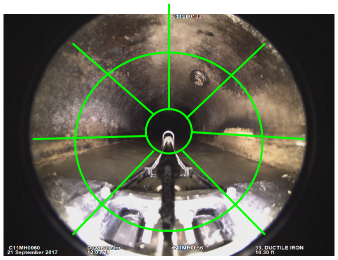
Image 1: In this example the inspector has identified a condition called RMJ. The model hear would recognize that and capture all the necessary information to go along with it such as clock from position, clock to position, percentage of defect, etc.

Image 2: In this example the inspector has identified a condition called FC. The model hear would recognize that and capture all the necessary information to go along with it (eg. Crack circumferential from 2 o'clock to 4 o'clock)

Table 1: For every condition there are fields that are required. This table gives the rules and names for each of these fields. This would be the format of every identified defect condition.

11. Pipe – Inner diameter identification

This model would identify the pipe inner diameter so that in the future the shape of the pipe could accurately be measured. This would allow to identify areas of pipe deformation such as can be seen in image 2 below, where the pipe is deformed horizontally. When matched with the OCR Pipe Size Analysis this would also allow us to know what the actual diameter is so it could be used to help with the Distance to condition.



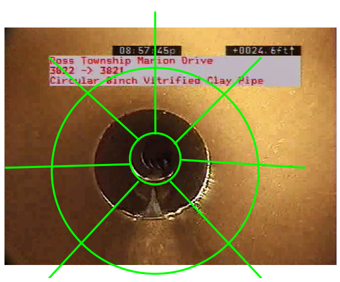


Image 1: Not sure how we'd identify the edges of the ID so just wanted to show a 3D representation of how a pipe ID will get smaller as you get more towards the center of the screen. With an inspection done with a fisheye we maybe could use the blank line as the segmentation point.

Image 2: Another example of how I'm not sure how we'd identify the edges of the ID so just wanted to show a 3D representation of how a pipe ID will get smaller as you get more towards the center of the

Image 3: In this example you can see how the pipe has deformed horizontally and no longer has the same diameter circumferentially. This is a severe condition that can lead to catastrophic pipe failure if left without further

12. Pipe – Distance to condition

This model relates to both the OCR Pipe Size Analysis and Pipe Inner Diameter Identification. The thought being that if we know the inner diameter and the size, then we could use that to potentially identify how far down the pipe a condition is as the inspection video begins to see it. Objective: For each defect condition identified this model would also provide an assessment of the distance from the camera lens the condition exists.

Image 1-3: From left to right you see how the given root goes from in the distance to closer to finally where the camera is capturing just the last bit of root and is most likely at the point of the condition. Using the 3rd image as ground truth, so the condition is at 123.9 ft, then when looking at image 1 this model would output 2.7 (123.9-120.2) if it worked accurately.

13. Wind – damage detection and identification

We have over 55k images of wind turbines with nearly 25% of those having some level of damage. The objective of this model would be to build an object detection model that locates and classifies the damage on a wind turbine blade.

Image 1: In this example you can see a crack when you zoom in on the high definition image.

Image 2: Example of image with no damage, but has a lightning receptor.