

Artificial Intelligence and Its Role in Improving Automated Optical Inspection

Edward Pechin
ViTrox Americas, Inc
TX, USA
Edward.pechin@vitrox.com

ABSTRACT

Artificial Intelligence (AI) in manufacturing is becoming more necessary to stay competitive, especially in the United States where labor cost is increasing year over year. AI that can think, feel, and act in a way that is parallel to humans is becoming invaluable and is also in high demand. The evolution of AI programming will almost certainly surpass human capability to develop accurate programs quickly and without error. The need for this in our industry certainly makes this a priority for a lot of vendors. In my presentation I'd like to discuss AI and its usefulness and absolute necessity in developing and improving AOI programs.

The utilization of AI in AOI programming has become increasingly important in our industry. To be able to complete programs using Smart Programming to save time and free up Engineering resources is a necessity. The continued improvement and development of this AI to debug and Quality check programs will become an ever-important tool that can start to level the playing field for a lot of US manufacturers.

The ability to review and classify defects is another integral role AI can play in reducing labor costs. Being able to utilize AI that you can reliably count on to make decisions and be more efficient than human inspectors is incredibly powerful. AI Review that provides feedback toward the AOI machine to improve AOI programs can cut down and eventually eliminate false calls and program escapes. This in turn will dramatically improve yields and throughput for all manufacturing companies.

INTRODUCTION

One of the issues we run into when it comes to creating, maintaining, and ensuring quality of AOI Programs is the amount of time it takes to do so. In the United States where we have a lot less resources, and in most cases are overwhelmingly becoming high mix low volume manufacturers; We are all tasked with supporting multiple processes at once. Because of this we are always on the run. Doing our best to put out fires here and there and keeping production moving along. The AOI machine is one of the most important and powerful inspection machines in our process. They can help to raise first pass yields, identify broken processes, and since they are inline, they are able to detect a defect as close to immediately as we have. So why do we rush such an important process where the quality of the program means everything? There are so many answers to such a question, and all of us have our own answers you can

almost hear while reading this, coming from your various site management teams.

In a perfect world we would be given the proper amount of time to create our programs, quality check each part one by one ensuring they are all setup properly to capture defects on the board, all while limiting false calls. In the real world, this is not the case. When creating a new AOI program, a Pilot run let's say, we try to get things ready ahead of time as best we can, but a lot of that is guess work. Depending on your AOI vendor you may have some existing database, or a database you can copy from to have a good start, but there are always new parts to train. So, we try to get this program spit out as quickly as possible. Doing our best in the limited time we have to ensure we are creating a program as robust as possible. Once a board hits our machine, we get the false calls to an acceptable level for operations, and we are calling it done and moving onto the next machine or issue. The next time we may be looking into fixing this program is if false call rates go up, or the worst-case scenarios, when we start to get program escapes. The constant back and forth, patching and band aiding programs is not the best way to do things, but it is the reality we are all faced with.

So, what do we do to correct this? We all do our best, I am sure. Maybe some of us are blessed with an excellent AOI programmer or team and the amount of impact is minimal. So, what do you do when they leave, and go find a better job? Or better yet, get to call it a career and retire? How do we pass down countless years of knowledge? Well, we really can't and we really don't. There has been a lot of aging out lately in our industry, employees who have tons of knowledge stepping down and going into retirement. How do we minimize this type of impact on our industry? Well Artificial Intelligence may be the answer to a lot of these problems.

When we talk about AI fortunately or unfortunately depending on how you see things, we are not talking about Skynet. So robots are not yet going to take over your factory, stealing your jobs in the process, then turning their focus to taking over the world. I am also not talking about Lights Out Manufacturing where we can turn off the lights and walk out of the factory and everything is going to run and maintain itself, most importantly never stopping or failing. Now, most of that is some higher up's fantastical dream of what AI is. Maybe one day it will be a reality, but today we are not quite there yet. So, how can AI help us out today? In particular for this presentation, how can it help us with the AOI process?

AI AUTO PROGRAMMING

When I first started working on AOI programming, everything was manual. For example, to create an SOIC algorithm we were setting the body size, number of leads, the pitch, setting the polarity, etc. With 2D programming we had to tinker with lighting angles to achieve a good contrast to be able to lock our algorithm onto the part. This was incredibly time consuming. 3D AOI programming did help with this but now we have some new opportunities to catch defects. Lead coplanarity, raised components, these are great tools that can be used to capture defects but again time consuming on the setup side.

That brings us to one of the first use cases for AI I will speak about today. What if we can use AI to do the initial training of a Pilot or NPI run? How many components could we have trained autonomously and how much time could that save us? As of today we are seeing that 96% of all SMT components packages can be trained automatically.



Figure 1. Auto Programming

As far as time saved? Depending on the complexity of the board we are seeing anywhere from 80-90% improvement in initial programming time. In almost all cases, AI can help to train over 90% of your components in minutes. Now comparing that to the hours it took before this is a significant improvement. All of the items I mentioned earlier, the manually setting up leads, pitch, polarity, coplanarity, OCV/OCR, etc. All done autonomously, all done at the click of a button.

Board Complexity (Component Count)	Programming Time (minutes)		Improvement
	Conventional Programming	A.I. Auto Programming	
High (>1500)	~180-240	~30 - 40	up to ~88%
Medium (500-1500)	~60-120	~20 - 30	up to ~83%
Low (<500)	~60	< 10	up to ~88%



Up to ~90% Programming Time Saving that covers >90% components populated



Suitable for High Mix Low Volume with lots of NPI

Figure 2. Auto Programming, Key Benefits

What other benefit can this AI auto programming have for us? Well, now we have newer engineers without a ton of knowledge or experience who can start a brand new AOI program. Being able to spread out your resources regardless of experience is a huge help to all of us. No longer would you

have to struggle when your main AOI programmer is on vacation or stuck working on another process. Now any one of your coworkers can help out and start up a pilot program regardless of experience. This is one of the most powerful tools and use cases for AI in AOI programming. Just in my experience, I cannot tell you the number of hours I spent programming new Pilot runs and the potential this AI has for saving us time and limiting stress on our resources.

AI RECIPE QUALITY CHECKS

Another way AI can assist us in our AOI process is Recipe Quality Checks. AOI Programmers, Engineers, and Technicians all make mistakes. We try to “Buddy” check one another, try to limit human error, come up with processes which ensure redundancy and still from time to time we fail. Escapes happen, usually it is a small human error that creates costly defects being passed beyond the AOI machine. Sure we can fix the AOI program after the fact. After it is flagged by the next downstream inspection process, after several defects have already been run through the SMT line. If only we had time to check each program before it is loaded and while it is running, especially directly after someone makes a tweak to the program. We all know we are never given this time and in most cases we do not even have that time to give if we were given the opportunity.

Where can AI help us in quality checking our programs? As Engineers or Technicians make changes to the recipe, AI can double check that our settings are all within IPC class standards. If an engineer or technician accidentally fat fingers a setting or makes a mistake, this AI can instantly flag this as an issue and will automatically fix parameters.



Figure 3. AI Quality Check, IPC Class Std.

Prior to loading a recipe AI can check for any untrained components in your recipe to be certain all parts are being inspected. If any of these parts are untrained, you will be flagged for this prior to running the recipe.

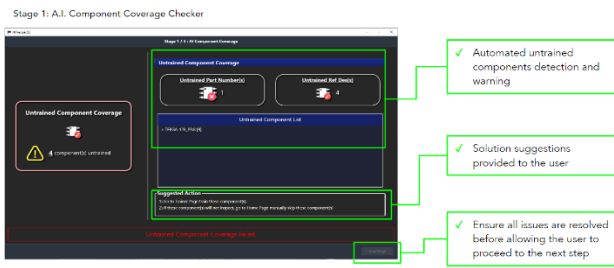


Figure 4. AI Quality Check, Untrained Part Check

Running an AI algorithm threshold optimization checker. Confirming that algorithm settings are enabled and optimized to the maximum ability to capture defects. Is polarity turned on, are joint and lead settings setup, are any of my thresholds set too loose and maybe causing escapes? The AI can help us to check these settings. After reviewing the report by AI, users can select to allow the AI to optimize these settings for them automatically.

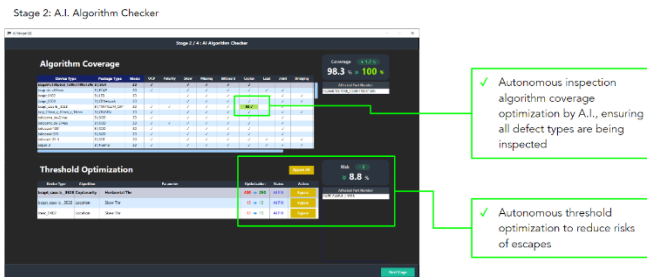


Figure 5. AI Quality Check, Optimization Check

Performing real time negative checks for OCV and polarity settings. This virtual check will detect and warn of potential OCV and Polarity escapes. It will also be able to guide users on how to resolve them.



Figure 6. AI Quality Check, Polarity and OCV Defect Simulation

The AI could also run algorithm validations such as missing components or bad joint simulations to ensure thresholds are set to detect these types of defects.

Stage 3: A.I. Algorithm Validation

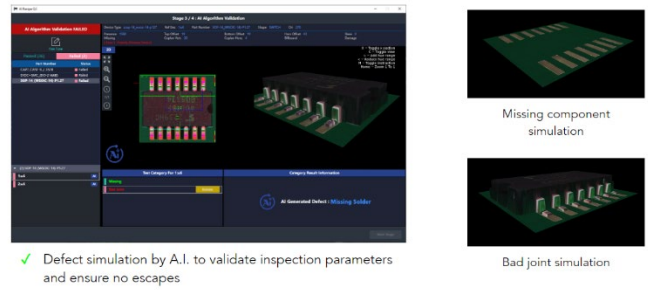


Figure 7. AI Quality Check, Defect Simulation

Not only can AI quality check our program and be almost 100% accurate in doing so, but it can also do it in a fraction of the time it would take a human to do so. In most cases we are finding that AI can do all of these checks and corrections >70% faster than an AOI programmer. Again, this is another great use case for AI and we can all see the instant value it can add to our process.

Key Benefits

Board Complexity (Component Count)	Time Taken to Validate Inspection Recipe Quality (minutes)		Improvement
	Without A.I.	With A.I.	
High (>1500)	45	10	77.78%
Medium (500 - 1000)	30	7	76.67%
Low (<500)	20	6	70.00%



Up to ~80% recipe validation time saving that covers >90% components populated



Multiple A.I. validation gates for escapes prevention, including IPC-compliance validation

Figure 8. AI Quality Check, Key Benefits

AI FINE TUNING

On established programs, a lot of the time spent fine tuning comes from vendor changes on the SMT lines. With vendor changes, part colors, polarity, OCV markings, and sometimes body sizes change and need to be tweaked. This constant tweaking does take a lot of our time, making it to where we are going back and forth between machines and processes like a ping pong ball. What if AI can help us with these simple changes?

How will AI know that a vendor change is exactly that and not a defect? I guess there are a lot of different methods that can be deployed here but what if we leveraged an already existing partner in the AOI process?

Every single AOI machine has an inspector. Sure, sometimes we have multiple machines being monitored by one inspector, but eventually it boils down to every AOI machine being monitored by an operator. These operators are ultimately responsible for going through all of the false calls. Making decisions on true defects vs false defects. When they are hit with a vendor change and start to false call the same

part number and reference designators AI can flag this part and can suggest parameter changes.

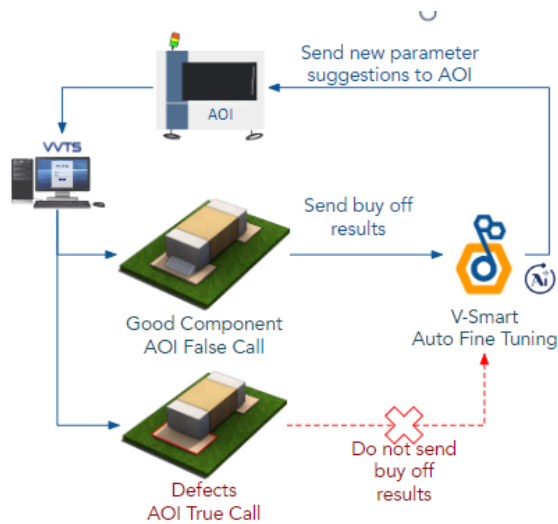


Figure 9. AI Fine Tune

The AI would be able to use all the production images collected overtime to determine the optimized parameters for each part. Along with leveraging the classification coming from the review station operators the AI would be able to make suggestions to improve false calls and automatically fine tune.

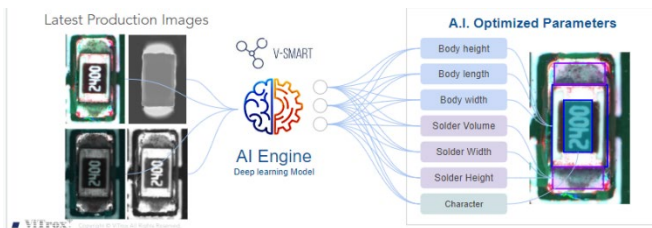


Figure 10. AI Fine Tune, Template Creation

We wouldn't want the AI unknowingly changing our programs, so we could have the AI notify us that this part number is experiencing high false call rates. It could then show us its suggested optimized parameters and again with one click of the button Approve or Reject the changes. Once we approve them AI will implement them immediately to where the false calls will instantly be reduced.



Figure 11. AI Fine Tune, Approval

AI Auto Fine-Tuning can cover 95% of SMT components and would reduce the amount of time needed for fine tuning over 70%. This again would be another huge improvement for us in terms of time, productivity, and quality.

A.I. Auto Fine-Tuning Performance

Fine Tuning Performance			Fine Tuning Effort Improvement			
Total No. of False Calls	Total Fine-Tuned by A.I.	Coverage	Fine Tuning Time		Time Reduced with A.I.	Improvement
86	82	95.35%	Without A.I.	With A.I.	8 mins 3 secs	-70%
			11 mins 33 secs	3 mins 30 secs		



Figure 12. AI Fine Tune, Performance Study

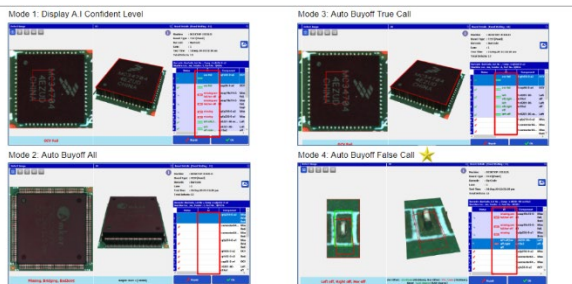
AI DEFECT REVIEW

The AOI review process is incredibly critical to our overall yields and production quality. Sure, a perfect AOI program can spot almost every defect coming off an SMT line, but can your inspection operator spot it? It is incredibly difficult to do so when we are forcing them to look at multiple false calls before finding the defect. The repetitive nature of clicking through false calls while also thinking about various other human endeavors makes it very likely that even our best operators will miss a defect from time to time. You also have the variation between your different operators, some can spot a defect better than others. Some are more attentive, have more experience, etc. All of this leads to a very unstable review process.

If we could utilize AI to perform auto judgement it would significantly cut the workload on the inspection operators. We could also reduce the number of escapes that pass through our AOI process due to human error. The fact that the AI will always be consistent we eliminate the constant variation between different operators with different levels of experience or knowledge.

So how can we implement this type of AI? There are a few different use cases that can be implemented based on various customer or site needs. We can have AI be an assistant to your operators, displaying a confidence level the AI has classified the part as showing either good or bad. We can have the AI buyoff the false and true calls automatically. We can also allow your operator to have control of the final judgement, or be password protected and unable to change what the AI classification was.

Use Models



Different running mode with or without A.I. Auto Judge

Figure 13. AI Defect Review, Use Case Models

In most cases, customer sites would probably start by having it setup as an assistance to the operators. Helping the operators to determine between the true and false calls. Mainly this would be done because most people don't want to blindly trust the AI judgement without first testing it out. From here as your confidence level grows you can progress to allowing the AI to perform judgement on just false calls, and soon enough to allow the AI to do all the calls whether True or False. We would certainly like to monitor the AI to make sure it is judging components properly. To do so we would have the AI send all of its classification results to a server-based application where we can monitor this and if need be change the results. If we find that AI judged a True call that was later determined to be false, we could correct it, and in doing this review, help to improve the AI's learning in real time.

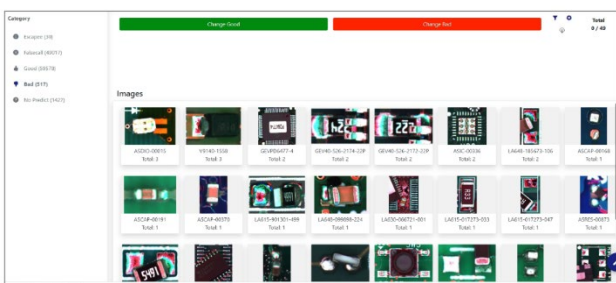


Figure 14. AI Defect Review, Verification

In a recent study involving almost 54k AOI calls, the AI was able to handle 64.7% of them, close to 35k. It was able to do this with a .09% escape rate. In most cases we believe the workload impact has been anywhere from 50-70% reduction in operator judgement.

A.I Coverage

The A.I Coverage up to ~64% which able to cover **Missing, Billboard, Tombstone, Flipped, Offset, Skew, Obvious Defect and OCV.**

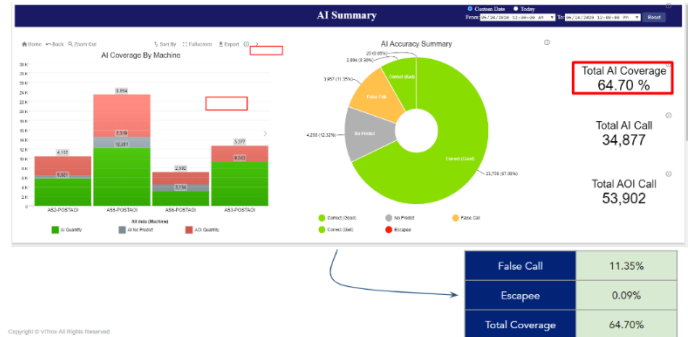


Figure 15. AI Defect Review, Case study

CONCLUSION

AI is or will become necessary in the high mix low volume production scenario that most of us are tasked with maintaining. With the help of AI we hope to ease the workload of engineering and operation teams while also improving labor cost, efficiency, and product quality.

After reviewing the four AI use cases presented in this paper, it is very apparent the overwhelming benefit AI can have with our AOI process and programs. Auto programming to help with tedious pilot program part training. AI Quality checks to double check our programs and make sure they have 100% coverage and are robust as possible. AI Fine tuning for those cumbersome vendor changes and slight variations in part sizes. Finally, AI Review to help our inspection operators navigate through the tumultuous amount of noise to find the one, or more, true defect.

Are any of these AI use cases perfected? As an AOI programmer, do I need to go out and find a new occupation or career? Not yet, and to be honest not by a long shot. Today, AI should be used more like a tool. It won't do the job for us but it can certainly assist us in our daily tasks. Usually, you the end user will help to perfect each use of AI. Whether it is related to AOI or any other process, AI needs human interaction to improve and evolve. Companies like Vitrox have entire teams dedicated to the R&D of AI so it will continue to improve daily. It will have a huge place in almost every industry in our future. Why not utilize it in our industry today?