

COGNEX DEEP LEARNING DEPLOYMENT GUIDE

Cognex Best Practices

COGNEX

COGNEX DEEP LEARNING DEPLOYMENT GUIDE

Cognex Best Practices



CONTENTS

Introduction **3**

How Is Deep Learning Different from Traditional Machine Vision? **3**

Return on Investment (ROI) **5**

Direct ROI 5

Indirect ROI 5

What Makes a Good First Project? **6**

Project Scope 6

Integration in Production 6

Acceptance Criteria 7

Development Resources **8**

Vision Developer 8

Quality Expert 8

Image Labeler 8

Data Collector 8

Project Phases **9**

Overview 9

Prototyping 10

Image Data Collection 11

Optimization 12

Validation and Deployment 13

Conclusion **15**



INTRODUCTION

Deep Learning is enabling new opportunities in automated manufacturing and inspection. A successful project can generate cost savings, as well as yield improvements and a better understanding of your own manufacturing process. But misjudging your first project can waste time and money and lead you to draw the wrong conclusions about what deep learning can and cannot do for you.

Many deep learning systems perform well in the lab but struggle when installed in the production environment. Oftentimes, user frustration stems from the underappreciated differences between deep learning solutions and traditional machine vision systems. This paper will help guide you through the key project stages, from selecting a good first project, to optimizing your application and validating it on the factory floor.

HOW IS DEEP LEARNING DIFFERENT FROM TRADITIONAL MACHINE VISION?

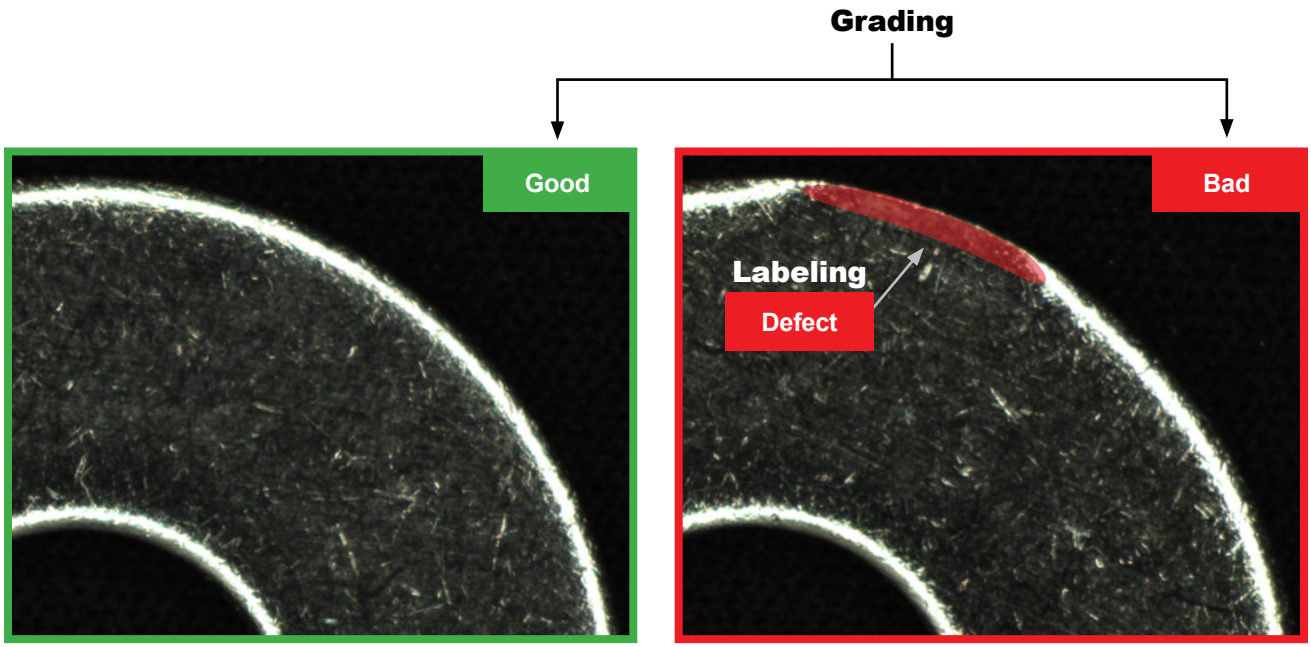


Neural networks are trained by example from images of good and bad parts. With their human-like judgment, they're especially well-suited to ambiguous problems such as cosmetic flaw detection and final assembly verification. They differ from classical machine vision approaches, which rely on rigid rule-based decisions using image processing filters, gauging tools, and other traditional vision algorithms.

Unlike traditional machine vision solutions, deep learning neural network applications are not programmed explicitly. Instead, they're trained by example. The cornerstone of your application will be the images from which you train your neural networks. A well-trained neural network requires a comprehensive set of training images that represent the range of defects and acceptable part variations that it needs to handle in production.

Qualifying a deep learning vision solution is an iterative process that requires the system to be installed on a production line. You'll need to train and validate it using images acquired under manufacturing lighting and part presentation conditions. This is essential, and unlike traditional machine vision, it must be done during the development phase—it cannot wait until factory acceptance testing. And since high quality manufacturing processes only produce a small percentage of defective parts, any inspection application will need time to capture a sufficient number of examples from the thousands of good products coming off the line.

In addition to collecting representative images, it will be important to grade and label them correctly. **Grading** refers to determining the *overall ground truth* as to whether a part is good or bad, or which type of defect is present. **Labeling** is the act of identifying *specific features* in the image, such as defect or part locations, for training the neural network. Carefully grading and labeling the images is critical to a successful project, so a quality expert must be part of the process from the very beginning.



In this example, grading refers to the overall "good / bad" decision for each part, while "labeling" is the marking of specific defect pixels in the image.

Once development is complete, the system will need to be tested statistically before entering production. Unlike traditional vision, it's not possible to analyze a specific image and understand precisely why the neural network reached its conclusion. The only way to qualify it is by measuring its performance over statistically valid data sets. In this sense, it's no different than a manual inspection process.



RETURN ON INVESTMENT (ROI)

While 100% automation accuracy is of course the perfect result, a realistic project payback is normally met by either:

- Reducing costs, while maintaining equivalent results to your current approach; or
- Improving results, while holding costs constant

Yield and throughput are important when computing payback, not just result accuracy. If you're using a manual process today, establishing a baseline for comparison requires tracking over enough time to generate statistically valid results. Human judgment is often superior to automated solutions, but most operators can only maintain their concentration for 15–20 minutes at a time. Another common source of human inconsistency is labor turnover, which can result in varying skill levels between operators.

DIRECT ROI

Computing direct ROI is a straightforward exercise. It compares the costs of the automation solution to your current approach (frequently manual inspection), while accounting for changes in throughput and yield.

Deep learning automation costs include:

- **Software and hardware expenses** such as licenses, cameras, lights, PC, and handling system. Most of this cost is incurred up-front but can be amortized over an equipment lifetime of 4-5 years.
- **Development costs** for engineering personnel.
- **IT data collection** of images, ground truth records, and manual inspection results. The infrastructure is generally lightweight, and should be re-usable on future deep learning projects.

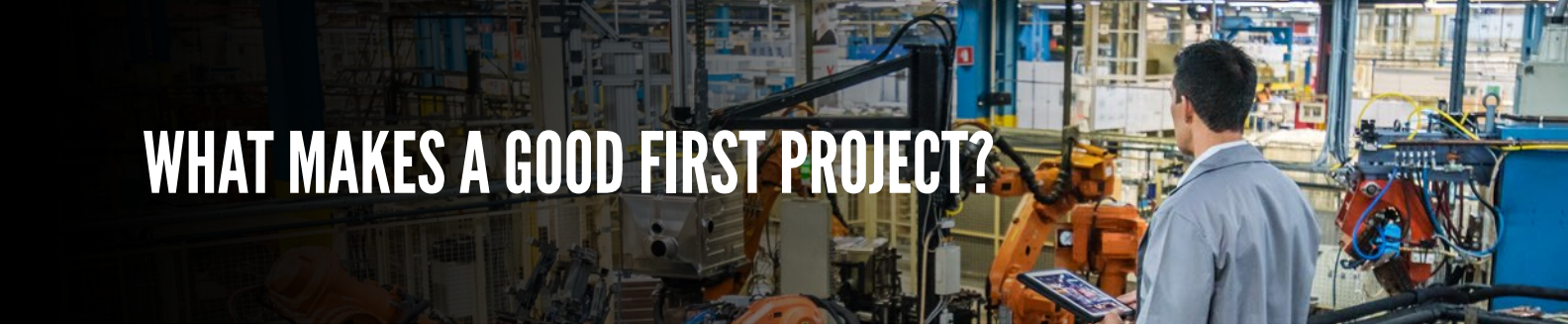
Manual inspection is dominated by labor costs, incurred yearly, and include staff turnover and re-training expenses.

INDIRECT ROI

Deep learning machine vision offers additional benefits beyond those in the direct financial ROI calculation:

- **Traceability:** Documenting inspection results for your end customer, as well as the ability to retroactively check inspection images and decisions in the event of future failures.
- **Continuous Improvement:** Logging images and decisions makes it possible to improve training, producing increasingly accurate results over time.
- **Upstream process control:** Once a final inspection station has been successfully automated, it's often possible to migrate the inspection steps upstream to in-line inspection. This cuts costs by allowing you to identify defects sooner, without adding value to bad parts.
- **Analytics:** Deep learning machine vision can be tied to your overall process improvement initiatives, such as correlating concrete vision data to other metrics such as process recipes, component suppliers, equipment differences, factory location, etc.

Even if you can't attribute a precise figure to these indirect benefits, it's important to include estimates to accurately justify your project. These also serve as placeholders for future opportunities.



WHAT MAKES A GOOD FIRST PROJECT?

Your initial deep learning project typically has two goals:

- Automating an inspection or verification step that is currently done manually or not at all
- Evaluating deep learning's broader utility for your future automation strategy

Choosing the right project can help you achieve both.

PROJECT SCOPE

Given that your first deep learning project will include a trial and error process and will inevitably involve mistakes, you should select one that is challenging enough to demonstrate the system's value, but realistic enough that it has a good probability of success. Many automation managers want to understand what deep learning can and cannot do for them. They review their proposals and start with the hardest, most ambitious challenge on the list, thinking *"if deep learning can solve this, it can solve anything."* While you can achieve a lot with this approach, it can also lead to delays and frustration for reasons that have nothing to do with deep learning. The project itself may be inherently too difficult or unstable. It may be cancelled if it fails to show rapid progress. And even if it eventually succeeds, it may steer you to the wrong conclusions.

We recommend starting with a project with clear payback that can't easily be solved with traditional vision, but which isn't so difficult that it never makes it into production. Focus on the core need, and defer other potential inspections until your understanding of deep learning grows.

We also suggest breaking the first project into discrete, manageable blocks. Since the "learning" part of deep learning takes place on the factory floor, you'll want to move out of the lab as soon as possible. Because deep learning involves a period of continuous improvement, you should aim to demonstrate consistent progress towards your final goal, rather than expecting an instantaneous solution.

INTEGRATION IN PRODUCTION

Cognex Deep Learning projects are typically installed in one of two places in your manufacturing line:

- At the end of the line, for final inspection or assembly verification
- In-line, to catch problems between manufacturing steps

There are advantages to each approach.

Final Inspection

Many factories include a manual quality check at the end of the production line, in which case you may already have explicit quality requirements in place. But the inspection task may be quite challenging, as it needs to account for the full range of potential defects that could appear in the final product.

One way to manage this is to start with a well-defined application like assembly verification. Assembly verification means confirming that all components in a package are present and correct, but without inspecting each component for defects, which can be much more challenging. Traditional vision systems often struggle with this application due to variations in lighting, perspective, and part appearance, but deep learning is uniquely well-suited for these issues. Once the assembly verification process is qualified, you can decide whether to add incremental inspection steps. Breaking the final inspection solution into smaller pieces allows you to simplify more complex applications.

In-Line Inspection

In-line inspection checks parts between two stations of the manufacturing process. Its scope is usually limited to the individual process step being performed. Because many automated manufacturing stations include a vision system for guidance or gauging, you may already have a camera in place and possibly a large collection of images from which to train a deep learning system.

In-line inspection is valuable because it eliminates bad product from your process before you add incremental value. It also offers the possibility for real-time process control, such as catching an empty glue dispenser or a misaligned part handler.

Throughput is the biggest challenge for in-line inspection. The automated solution needs to keep up with the production line to avoid becoming a bottleneck. Depending on the process speed, this may require more expensive vision processing hardware. A second challenge may be the lack of defined quality standards at this intermediate step.

ACCEPTANCE CRITERIA

Traditional vision application tasks such as alignment, gauging, and identification have clear answers: a part is found at a certain location, the gap between two edges is so many millimeters, or an ID code translates to a particular SKU. Deep learning applications are more subjective, so you'll need clear criteria to train and validate the system.

It's essential that your team agrees on the definition of good or bad parts. For a deep learning system, inconsistent training data is worse than no data at all. To avoid mis-training, you'll need to carefully identify and segregate any ambiguous images from your training sets. You'll also need to take these boundary cases into account when calculating your current production yields and ROI.

It's also important to ensure that the correct decision can be determined simply from an image of the part. Watch how the operators inspect parts on your production line. Do they pick them up and tilt them under the light? Do they brush off oil stains before re-positioning them in a tray? If so, it's likely that images alone aren't enough for the deep learning system to reach a correct decision. Have your quality experts examine images of the parts. Can they make the right determination? If not, you'll need to address your lighting and part handling, or consider another project. If a human can't reach the right decision from an image, neither can a deep learning-enabled vision system.



DEVELOPMENT RESOURCES

There are four core roles performed in a deep learning project. Though it's possible for one person to cover more than one role, you should be aware of the different requirements when selecting project staff. Because of it's focused tool design and simplified user interface, the Vision Developer does not need to be a neural network expert. Machine vision experience and a basic understanding of deep learning principles are sufficient.

VISION DEVELOPER

The developer implements the deep learning vision solution. His other key task is to optimize the lighting and image formation.

QUALITY EXPERT

The Quality Expert's primary role is to analyze images and grade them. Grading is the act of determining the overall *"ground truth"* for the part in the image: pass/fail, type of defect, etc. The Quality Expert also helps finalize the project acceptance criteria.

The grading role is critical. The vision system relies on clear and consistently graded images for training. The quality expert is also used to resolve "ties" when the vision system and manual inspector(s) disagree. It's important that the quality expert be trusted by all members of the team and those interacting with the team.

IMAGE LABELER

If grading is the act of assigning the ground truth result to a particular part, labeling is the interactive process of painting or otherwise indicating which pixels in the image define the defect or other features of interest. Labeling is a precise, detail-oriented process that needs to be done accurately and consistently on every image in the training set.

In the early prototyping phase, the Vision Developer or the Quality Expert may label the images. As the project accumulates larger image sets in production, you may choose to assign a separate individual to this role.

DATA COLLECTOR

A deep learning-enabled vision system is trained from images, grades, and labels. The data collector needs to record and organize this information along with metadata such as production date and time stamp, part type, manufacturing line, etc. Optimizing the system will involve testing it against these different data sets. This data organization will help you isolate problems that only appear under specific conditions, and ensure you validate your solution across the full range of required cases.

If a manual inspection procedure is already in place, the data collector should also record the decisions of the inspector(s) and associate them with the image sets. This data can be used when measuring yield, analyzing overkill, and performing cost/benefit analysis.



PROJECT PHASES

Deep learning projects can be organized into four common phases:

Prototyping

Understand your current process and determine if deep learning is a good candidate to solve it.

Key Steps:

- Determine application requirements.
- Acquire a small database of graded and labeled images.
- Build a proof of concept system to test the approach.

Image Data Collection

Integrate the system on your production line and begin gathering data.

Key Steps:

- Integrate camera and lighting on the production line.
- Begin logging image data and manual inspection results (if any).
- Establish ground truth data and label image sets.

Optimization

Improve the deep learning solution until it meets your performance target.

Key Steps:

- Run the deep learning system against your image data sets and in production.
- Compare the results to ground truth and to manual inspection (if used).
- Adjust the system and re-train as needed. Repeat.

Validation and Deployment

Qualify the solution and begin using it in production.

Key Steps:

- Pass factory acceptance tests and lock the “golden” configuration.
- Go “live” in production.

PROTOTYPING

Understand your current process and determine if deep learning is a good candidate to improve and/or automate it.

Key Steps:

- Determine application requirements.
- Acquire a small database of graded and labeled images.
- Build a proof of concept system to test the approach.

In this phase, you develop a proof of concept to determine whether deep learning is a viable candidate for your application. The initial part of this work is typically done in the lab, based on a small training image sample set.

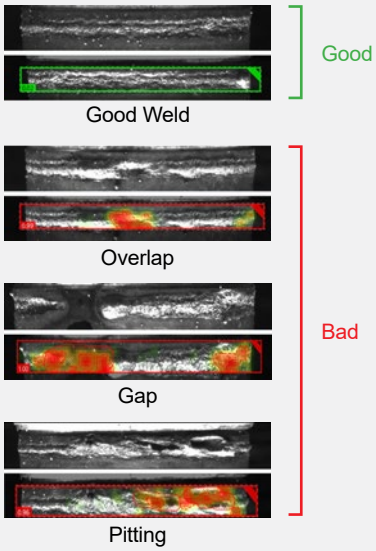
It's important to work with your Quality Expert to understand the core requirements, and to grade and label your initial image set. You need to confirm that correct decisions can be made by the Quality Expert from your images, as opposed to the physical parts. If the expert can't, then you should adjust your lighting or part presentation until the key features are clearly visible. If lighting changes still aren't enough for the Quality Expert to correctly grade the images, then the project may not be suited to deep learning.

As you build the prototype, experiment with the range of vision tests you might implement. Decide which are the most critical and which are the most practical, then determine those that you'll focus on first.

This phase normally concludes with a demonstration to project managers. If it appears promising, you should begin acquiring data from the factory floor.

Example: Project Scope and Prototyping Phase

Phase 1
Determine Good vs. Bad.
Ensure bad parts rejected or sent for manual rework.



Phase 2–Defer
Classify by defect type
Apply corrective action.

Defect Type	Severity	Root Cause
Good Weld	OK	N/A
Overlap	Borderline	Control Error
Gap	Defective	Control Error
Pitting	Defective	Weld Process

Limit the scope of the first project to Good/Bad defect detection.

Defer defect classification and process feedback until a later stage, after the first stage has been validated in production.

This will allow you to demonstrate progress faster, while learning lessons and gathering data that will make the second stage simpler to implement. For example, once the first phase is complete, you will already have an organized, graded database showing all necessary defect types, while your defect detection tools will have isolated the key regions for the classifier to focus on.

IMAGE DATA COLLECTION

Integrate the system on your production line and begin gathering data.

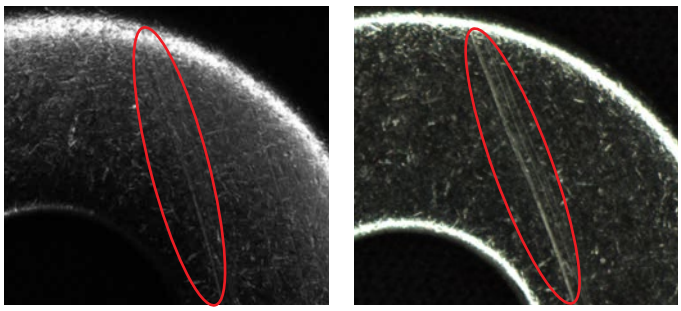
Key Steps:

- Integrate camera and lighting on the production line.
- Begin logging image data and manual inspection results (if any).
- Establish ground truth data and labelled image sets.

Image Formation

Once you've selected a candidate project, you should install a camera and lighting at the appropriate production step. Your IT resource can then begin logging images.

This is the best time to optimize your image formation system. Deep learning training is based on hundreds or thousands of images, each of which has been manually checked by your Quality Expert. It's incredibly frustrating to work on a project for a month or more, only to discard all the training data because the lighting system needs to change. It's much more important for deep learning projects to optimize the lighting early in the process than it is with traditional vision approaches. Once optimized, stabilize the image formation.



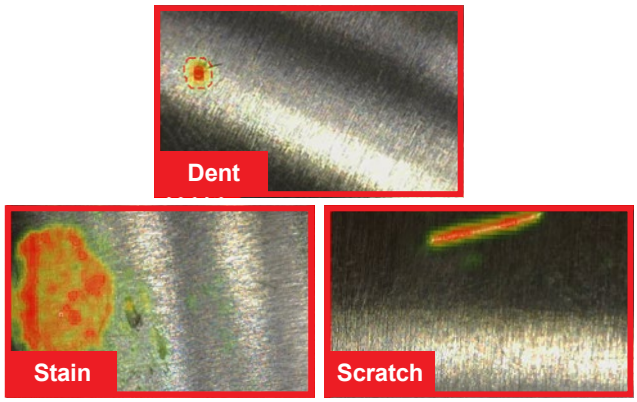
Proper lighting helps solve applications faster, and the resulting solutions are much more robust in production.

If you plan to deploy on multiple production lines, your image data sets must represent the variations in lighting and part presentation that the system needs to handle. Otherwise you'll be forced to develop and deploy unique solutions for each line. Fortunately, Cognex Deep Learning doesn't require images from every camera and lighting system. This will not only help with developing Deep Learning tools, but with general system maintenance. Normally, 3–4 lines are sufficient to reflect the range of part appearances the system must handle, as long as their image formation is consistent, and their c ontributions to the training set are proportional.

Start by optimizing the lighting on your first line. Once it's qualified, duplicate it on a few additional lines. One tip is to create a standard operating procedure (SOP) for image formation setup. This will not only help with developing Deep Learning tools, but with general system maintenance. A good SOP will cover camera mounting (working distance, field of view, orientation), focus, and brightness and contrast. This can be built from simple graphics to guide the mounting, and standard image sharpness and histogram functions.

Data Collection

Once your image formation and data recording are set up, you should begin gathering data. You should aim to capture images and results covering the range of expected product variations: lighting changes, component suppliers, and especially the varieties of defects that must be detected at the final inspection step. You should start this early, as it may take time to identify enough representative defects. This is especially true if you already have a high qualify manufacturing process in place.



As images are recorded, it's essential that they be graded and labelled consistently. If one image is described as "good," while another with a similar blemish is called "bad," the deep learning system will be confused. It won't know why the two images were classified differently, and so it will look for other variations in the image as the reason. These could be caused by shadows or hotspots, background noise, or other immaterial differences. And whatever it learns, it will apply in production. Similarly, it's fine for two different people to label image defects, as long as they apply the same standards. The Quality Expert should review all image grades and labels to ensure consistency, especially at the start of the project.

Be sure to record the manual inspection results along with the part images and the vision system's final decisions. This information will help you accurately measure the yield of the different approaches. And by comparing the automated and manual results, you'll be able to quickly identify those where the answers differ, which are the key to optimizing the system. Finally, it's also good to observe the inspection operators to confirm that no special part handling takes place as part of the manual inspection process.



OPTIMIZATION

Improve the deep learning solution until it meets your performance target.

Key Steps:

- Run the deep learning system against your image data sets and in production.
- Compare the results to ground truth and to manual inspection (if used).
- Adjust the system and re-train as needed. Repeat.

Optimization is frequently the longest phase of development. It usually takes weeks, and possibly several months. The project typically starts with rapid iterations of training. As new data is presented to the deep learning system, it is natural for performance to oscillate.

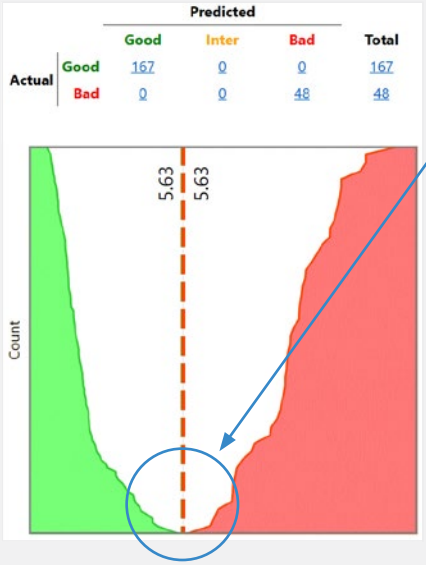
Performance should stabilize as the neural network is optimized and the frequency of new cases decreases.

The goal of this phase is to optimize the deep learning solution until it meets the necessary goals for deployment in production. The vision developer should be prepared to give regular progress updates to management. If senior managers aren't aware of the need for graded images covering the full range of production cases, they may be impatient and feel that the project is failing if it isn't ready to deploy immediately. In this case, it's important to have quantifiable data showing regular system improvement.

The optimization process requires careful training and observation of results. It's important to organize your data sets with all key variables, such as product type, production line, time of day, etc. That way you can dynamically create different test sets to qualify the system across a range of conditions. In some cases, the Quality Expert may need to review the graded images or fine-tune the labeling if they produce anomalous results.

Use this time to finalize the project scope and inspection criteria. Some inspections may prove too difficult and need to be deferred to a later phase. Other optional tests may turn out to be easier than planned and can be readily added.

Example: Optimization Phase

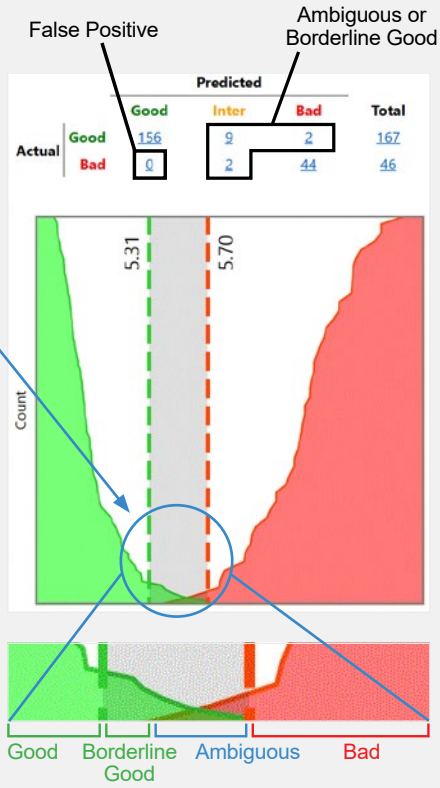


With inspection applications, your goal is to have a clear separation between good and bad parts.

Many times this is not practical, and there will be some confusion between border cases.

In this case, set your thresholds conservatively, to ensure no bad parts are incorrectly evaluated as good (false positive or underkill).

Any ambiguous results or borderline good images should be rejected, or tagged for secondary manual review.



VALIDATION AND DEPLOYMENT

Qualify the solution and begin using it in production.

Key Steps:

- Pass factory acceptance tests and lock the “golden” configuration.
- Go “live” in production.

Factory Acceptance Testing

Once the system is optimized and meets all off-line performance tests, a formal acceptance test is normally required. This may include both off-line and live tests including:

- A broad set of representative parts and defects
- Defect-specific validation against particular test cases
- A repeatability validation, where different images of the same part are tested multiple times

Statistical testing is the key to validating deep learning. Ultimately, the statistical testing process used will be a hybrid of those used for manual inspection and traditional vision. Remember, a realistic project goal is rarely to achieve 100% accuracy. Instead, it's to surpass current standards while holding costs constant, or to save costs while maintaining the current performance.

High precision geometric and image measurements aren't feasible with neural networks. Instead, deep learning validation is comparable to qualifying a manual inspection process. Even though you train human inspectors to perform certain quality checks, they can never explain their decisions in terms of specific grey level thresholds or blob areas. The best they can do is point to certain image attributes that affected the decision, which deep learning can also do.

Similarly, deep learning results are ultimately based to a defect probability map for the pixels in the image. The overall pass/fail for an image is usually tied to the maximum defect score for the image, and the repeatability and reliability of this pass/fail metric can be measured using standard Gage R&R methods. Additional results, such as the position and size of defect regions are derived from the defect probability map using user-defined thresholds, and it is generally not practical to use Gage R&R methods on these metrics.

A successful deep learning deployment lets you replace or augment your existing manual inspection process, improving reliability and consistency. And like any automated vision-based inspection, deep learning automatically provides you a wide range of opportunities to improve the validation, qualification, auditing, and tracking of your inspection process.

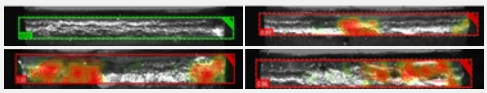
Two-Tiered Inspection

A common way to mitigate the uncertainty around a first deep learning project is to retain some level of manual inspection. A two-tiered inspection approach can yield good results and doesn't involve a duplication of effort. Instead, deep learning is used to scan 100% of outgoing parts. The vast majority are classified as unambiguously good or bad and handled appropriately. But a small portion fall into a grey area of mathematically lower confidence. These parts (or their images) are sent to a small team of manual inspectors whose sole job is to make the final decision.

This approach can be used during initial project qualification, until the system is fully vetted, or it can be maintained indefinitely. It offers several benefits compared to both an exclusively manual process as well to a fully automated solution:

- The **higher yield, reliability and cost savings** of an automated solution, with the **confidence of human judgment.**
- **Full engagement of manual inspectors.** In most manual inspection, an overwhelming majority of parts are good, and inspectors become easily bored. But in a two-tiered system, every part presented is one that needs their judgment. And since the frequency of these reviews is much lower, the inspectors have more time to decide the right answer instead of making snap judgments as parts flash by.
- **Feedback for deep learning improvement.** The system logs every image sent to the human inspectors, along with their decisions. This information can be retained to use for incremental training improvements.

Example: Validation and Deployment Phase



Validation data set collection

- Period: April 4 - May 27
- Lines: 1 - 4
- Shifts: 1, 2
- Coatings: Nickel, Silicon Carbide
- Defects: Overlap (313); Gap (147); Pitting (423)

Image data should be organized so each key variable can be independently tested

Ground Truth Grade	Deep Learning Decision	Count	Percent	
Good	Good	4,092	99.39%	Pass
	Intermediate	14	0.34%	Reject or Review
	Bad	11	0.27%	
	Total	4,117	100.00%	
Bad	Good	0	0.00%	No False Positives
	Intermediate	3	0.34%	Reject or Review
	Bad	880	99.66%	Reject
	Total	883	100.00%	
Grand Total		5,000		

Validation data sets are much larger than those used for training. They should include sufficient examples of acceptable variations as well as all defect types.

In addition, they should span all key production attributes that can affect part appearance: production facility, part type, production line, time of day, etc.

When integrating into production, review overkill and underkill rates and inspection times. These should be compared to the existing manual or automated procedures to determine ROI and setting pass/fail thresholds. Pay special attention to underkill (false positive) and the total associated costs in product recall and safety.

With fully automated inspection, reject any ambiguous parts. With a two-tier inspection, these can be directed to a single inspector who determines the correct decision. Images of the ambiguous parts and the resulting decision can used later for continuous improvement of the deep learning system.



CONCLUSION

Deep learning machine vision is opening automation opportunities across a wide range of industries. Inspecting surface defects, sorting variable parts, checking final assemblies, grading product quality, reading challenging text are just a few of the dozens of new applications which deep learning-enabled vision systems can ably handle, and no modern manufacturer can afford to ignore them.

But manufacturing lines aren't science labs, and plant managers rightly hesitate to risk their existing qualified processes in favor of a new technology's future rewards. By following the advice outlined in this paper, factories and manufacturers new to deep learning will find the guidance necessary to help them avoid the missteps, lost time, and organizational skepticism that can compromise early installations. From initial project selection all the way to final qualification, Cognex Deep Learning users can feel confident knowing that that their deployments are on the track to success.

COGNEX

Companies around the world rely on Cognex vision and barcode reading solutions to optimize quality, drive down costs and control traceability.

Corporate Headquarters **One Vision Drive Natick, MA 01760 USA**

Regional Sales Offices

Americas

North America	+1 844-999-2469
Brazil	+55 11 4210 3919
Mexico	+800 733 4116

Europe

Austria	+49 721 958 8052
Belgium	+32 289 370 75
France	+33 1 7654 9318
Germany	+49 721 958 8052
Hungary	+36 800 80291
Ireland	+44 121 29 65 163
Italy	+39 02 3057 8196
Netherlands	+31 207 941 398
Poland	+48 717 121 086
Spain	+34 93 299 28 14
Sweden	+46 21 14 55 88
Switzerland	+41 445 788 877
Turkey	+90 216 900 1696
United Kingdom	+44 121 29 65 163

Asia

China	+86 21 6208 1133
India	+9120 4014 7840
Japan	+81 3 5977 5400
Korea	+82 2 530 9047
Malaysia	+6019 916 5532
Singapore	+65 632 55 700
Taiwan	+886 3 578 0060
Thailand	+66 88 7978924
Vietnam	+84 2444 583358

© Copyright 2021, Cognex Corporation.
All information in this document is subject to change without notice. All Rights Reserved. Cognex and In-Sight are registered trademarks of Cognex Corporation. All other trademarks are property of their respective owners. Lit. No. CDLDG-05-2021 1000003311 01

www.cognex.com