

A Comprehensive Guide to OCR with RPA and Document Understanding



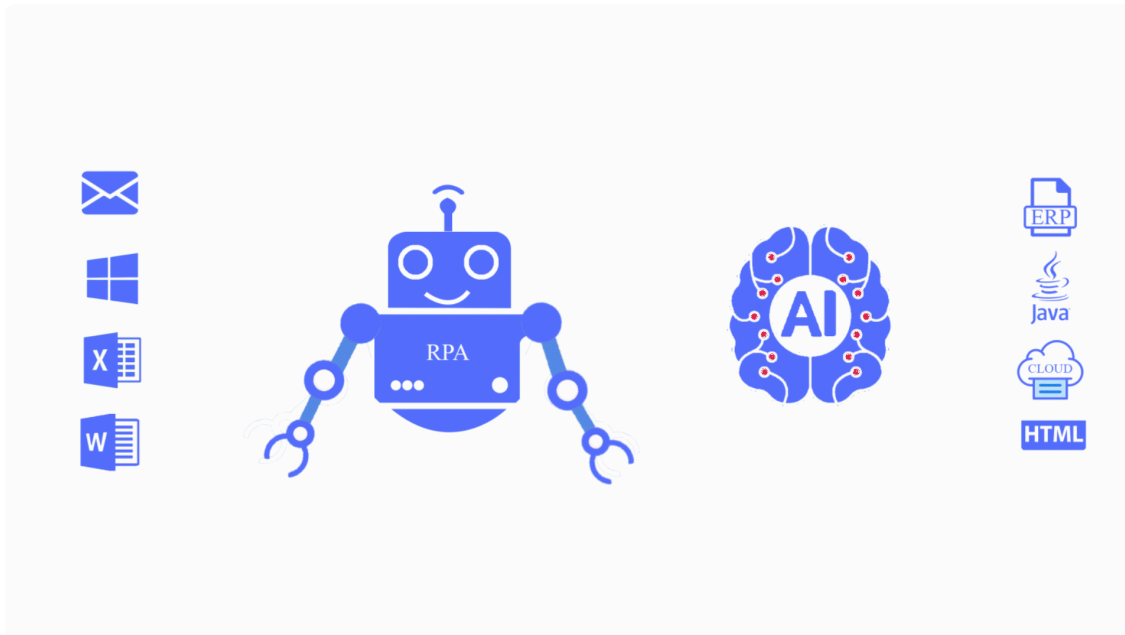
by **Vihar Kurama** 2 months ago

16 MIN READ

Last update: Jan, 2021.

This blog is a comprehensive overview of using OCR with any RPA tool for automating your document workflows. We explore

how the latest machine learning based OCR technologies don't require rules or template setup.



RPAs or Robotic Process automation are software tools aimed at eliminating repetitive business tasks. More CIOs are turning towards them to reduce costs and helping employees focus on higher value business work. Examples include responding to comments on websites or customer order processing. **Slightly more complex tasks include handling documents like handwritten forms and invoices - these typically need to be moved from one legacy system to the other - say your email client to your SAP ERP system where you need to extract data. This is the problematic part.**

Most OCR tools that capture data from these documents are **template based (say Abbyy Flexicapture) and don't scale well on semi-structured documents.** There are newer generation machine learning based solutions that typically provide API integrations that can capture key-value pairs from documents - enterprise systems typically are legacy and not open to integrate

with external APIs. On the other side, RPAs are built to handle these legacy system workflows like ingesting documents from folders and entering results into ERPs or CRMs.

As Robotic Process Automation (RPA) and ML are evolving towards hyper automation, we can make use of software bots in conjunction with ML to handle complex tasks such as Document Classification, Extraction, and Optical Character Recognition. In a recent study, it was said that by automating only 29% of functions for a task using RPAs, finance departments alone save more than 25,000 hours of rework caused by human errors at the cost of \$878,000 per year for an organization with 40 full-time accounting staff [1]. In this blog, we'll be learning about using OCRs with RPAs and deep dive into document understanding workflows. Below are the table of contents.

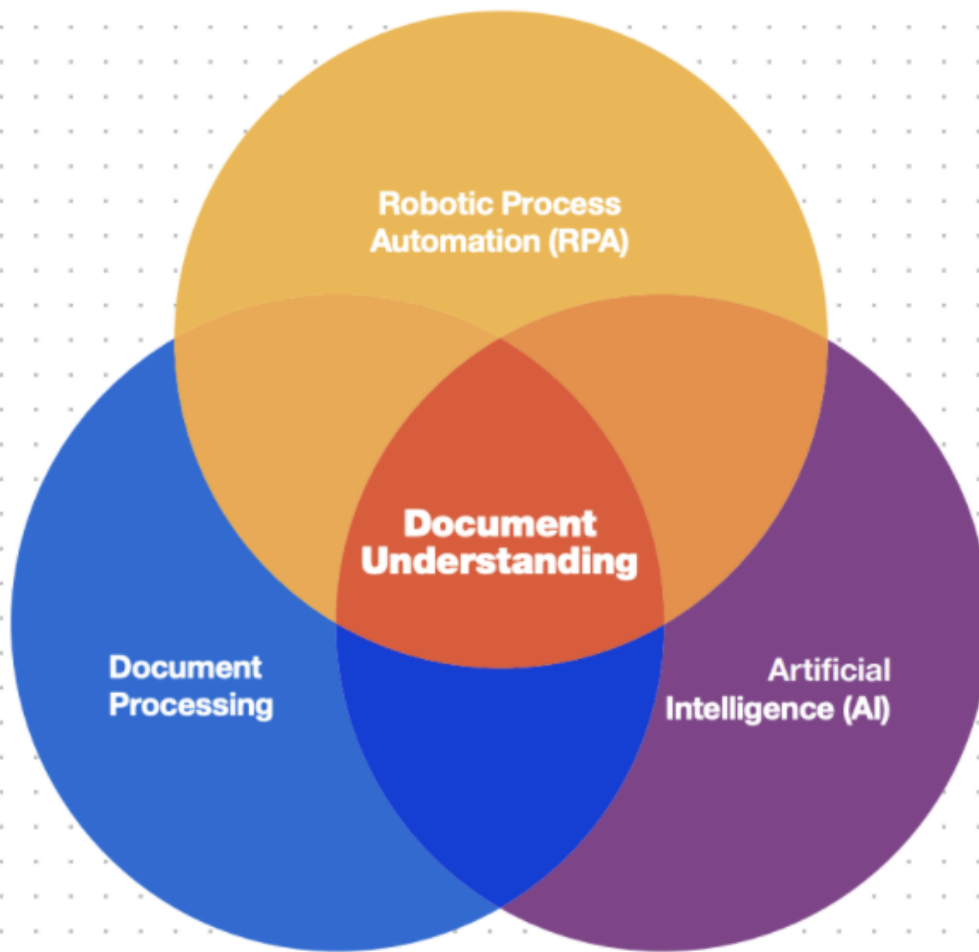
- Definitions and Overview
- How Robots can learn to understand the documents with OCR and ML
- How can companies integrate intelligent OCR with RPA to improve workflows
- OCR Challenges faced by RPA developers
- Pipeline for Document Understanding Workflow
- Benefits of integrating robots with Document Understanding
- Enter Nanonets
- Nanonets with UiPath

Definitions and Overview

RPA, in general, is a technology that helps automate administrative tasks via software-hardware bots. These bots take advantage of user interfaces; to capture the data and manipulate applications as humans do. For example, an RPA can look at a

series of tasks taken in a GUI, say moving cursors, connect to APIs, copy-pastes the data, and formulates the same sequence of actions in an RPA wireframe that translates to code. Further, these tasks can be performed without human intervention in the future. Optical Character Recognition (OCR) is a crucial feature of any functional robotic process automation (RPA) solution. This technology is used to read and extract text from different sources like images or pdfs into a digital format without manually capturing it.

On the other hand, **Document understanding is the term used to automatically describe reading, interpreting, and acting on document data.** The most important in this process is software bots itself perform all the tasks. These bots leverage the power of Artificial Intelligence and Machine Learning to understand documents as digital assistants. This way, we can say that document understanding emerges at the intersection of document processing, AI, and RPA.



Document understanding emerges at the intersection of document processing, AI, and RPA. [Image Source: Ui Path Document Understanding [2]]

Looking for an AI based OCR solution that'll work seamlessly with your RPA? Give Nanonets™ a spin and put OCR based process on autopilot!

Get Started

How Robots can learn to understand the documents with OCR and ML

Before we deep dive into Document Understanding first, let's talk about the role of Robots for Document Understanding. These utterly invisible helpers make our life's much more comfortable. Unlike movies and series, these robots are not physical devices or artificial intelligence programs that sit at a desktop and push buttons to perform tasks. We can think of these as **digital assistants who are trained to process documents by reading and using applications as we do**. On the functional side, robots are good at improving the performance and efficiency of a process. Still, they being a standalone software, cannot evaluate the process and make cognitive decisions. However, **if machine learning is successfully integrated, robotics will become more dynamic and adaptive**. For example, robots utilized for document processing, data management, and other functions across the front and middle office will perform more intelligent actions, such as eliminating duplicate entries or solving unknown system exceptions in the process. **Further, the robots are trained to read, extract, interpret, and act upon data from the documents using artificial intelligence (AI).**

How can companies integrate intelligent OCR with RPA to improve workflows

Extracting document data is a crucial component for document understanding. In this section, we'll be discussing how we can integrate OCR with RPA or vice-versa. Firstly, we all knew that

there are different kinds of documents in terms of templates, style, formatting, and sometimes language. Hence we can't rely on a simple OCR technique to extract the data from these documents. To address this problem, we'll be using both **rule-based approaches and model-based approaches within OCR to handle data from different document structures**. Now we'll see how companies doing OCR can integrate RPAs in their existing system based on the type of documents.

Structured Documents: In this type of documents, the layouts and **templates are usually fixed and almost consistent**. For example, consider an organization that does KYC with Government-issued IDs like a passport or driving license. All these documents will be identical and have the same fields as ID Number, Name of the Person, Age, and few others at the same positions. But only the details vary. There may be few constraints like table overflowing or unfiled data.

Usually, the recommended approach uses a template or rule-based engine to extract the information for structured documents. These **can include regular expressions or simple position mapping and OCR**. Hence to integrate software robots to automate information extraction, we can either use pre-existing templates or create rules for our structured data. There's one disadvantage using the rule-based approach, **since it relies on fixed parts, even minor changes in form structure can cause rules to break down**.

Semi-Structured Documents: These documents **have the same information but are arranged in different positions**. For example, consider invoices containing 8-12 identical fields. In a few invoices, the merchant address can be located at the top, and in others, it can be found at the bottom. Typically these **rule-based approaches do not give high accuracies**; hence we bring in

machine learning and deep learning models into the picture for information extraction using OCR. Alternatively, in some cases, we can use **hybrid models involving both rules and ML models**. A few popular pre-trained models are FastRCNN, Attention OCR, Graph Convolutions for information extraction in documents. However, again these models have few drawbacks; hence we **measure the algorithm performance using metrics like accuracy or confidence score**. Because the model is learning patterns, rather than operating off concrete rules, it may make mistakes initially right after corrections. However, the solution to these drawbacks – **the more samples the ML model processes, the more patterns it learns to ensure accuracy**.

Unstructured Documents: RPA, today is unable to manage unstructured data directly, hence requiring robots first to extract and create structured data using OCR. Unlike structured and semi-structured documents, **unstructured data doesn't have a few key-values pairs**. For example, in a few invoices, we see a merchant address somewhere without any key name; similarly, we observe the same for other fields like date, invoice ID. For ML models to accurately process these, the robots need to learn how to translate written text into actionable data, like an email, phone number, address, etc. The model then will learn that 7- or 10-digit number patterns should be extracted as phone numbers and huge text containing five-digit codes and different nouns as text. To make these models more accurate, we can also use techniques from Natural Language Processing (NLP) like Named Entity Recognition and Word Embedding.

Overall for document understanding, it is first essential to understand the data and then implement OCR with RPAs. Next, rather than mapping out a process step-by-step, we can teach a

robot to "do as I do" by recording the process as it happens with powerful OCR capabilities as discussed above, by integrating rules and machine learning algorithms. The software robot follows your clicks and actions on the screen and then turns them into an editable workflow. If you're working entirely in local programs, that's as much as you'd need to know.

OCR Challenges faced by RPA developers

We've seen how we can integrate OCR with RPAs for different documents, but there are a few cases of challenges where the robots need to handle well. Let's discuss them now!

- **Weak or Inconsistent Data:** Data plays a crucial role in Document Understanding. In most of the cases, the documents are scanned using cameras where there is a chance of losing document formatting during text scanning (i.e., bold, italic & underline are not always recognized). Sometimes, the OCR might extract text in the wrong way leading to spelling errors, irregular paragraph breaks, which reduces the overall performance of robots. Hence handling all the missing values and capturing the data with higher precision is vital to achieving higher accuracy for OCR.
- **Incorrect Page Orientation in Documents:** Page Orientation and Skewness is also one of the common problems that lead to incorrect text correction of OCR. This usually occurs when the documents are incorrectly scanned during the data collection phase. To overcome this, we'll have to declare a few functions to robots like auto-fit to the page, auto-filter so that they can enable the increase in the quality of the scanned document and receiving correct data on output.
- **Integration Problems:** Not all RPA tools perform well on remote desktop environments – they cause crashes and critical problems in automation. What's more, the RPA developer needs to know which OCR solution will be the best for a specific case. Also, to work with specific automation

preprocessing steps!

tools, the RPA developer needs to choose only limited OCR technology created by Microsoft, Google. Hence integrating our custom algorithms and models is sometimes challenging.

- **All of the text is scrambled text:** For real-life use-cases, text captured by a generic OCR is all scrambled and has no meaningful information that the bots can use to perform significant operations. RPA developers need strong ML support to be able to build useful applications.

Pipeline for Document Understanding Workflow

In the previous sections, we've seen how bots help perform OCR for different types of Documents. But OCR is just a technique that converts images or other files into the text. Now, in this section, we'll be looking at Document Understanding workflow right from the beginning of collecting documents to finally saving them meaningful information into the desired format.



Basic RPA Workflows (Source)

1. **Ingest the document from a folder using your Bot:** This is the first step through achieving document understanding through bots. Here, we'll be fetching the document located either on a cloud platform (using an API) or from a local machine. In a few cases, if our documents are on web pages, we can automate scraping scripts through bots where they can fetch documents on a timely basis.
2. **Document Type:** After we fetch the data, it's essential to understand the type of document and the format with which they are saved in our systems, as sometimes, we receive data from different sources in various file formats such as PDF, PNG, and JPG. Not just the file types, sometimes

when the documents are scanned with phone cameras, a few challenging problems like image skewness, rotation, brightness, or low-resolution should also be handled.

Thereby, we'll have to make sure that bots **classify these documents into the structured, semi-structured, or unstructured category**, thus saving it in a generic format. The classification task is **achieved by comparing the documents with templates and analyzing features like fonts, language, presence of key-value pairs, tables, etc.**

3. **Extracting the Data with OCR:** All right, now that the bots arranged our documents into a generic format and classified them, it's time for us to **digitalize them using the OCR technique**. With this, we'll have the text, its location in co-ordinates from the images. This helps to **standardize the documents and data for the subsequent steps**. We also encounter a few when OCR software could not correctly distinguish between characters, such as 't' versus 'i,' or '0' versus 'O.' The very errors you want to dodge using OCR software can become new headaches when OCR technology is incapable of analyzing the nuances of a document based on its quality or original form. This is where Machine Learning comes into the picture, which we shall discuss in the next step.
4. **Leveraging ML/DL for Intelligent OCR using Bots:** After the data is digitized, the OCR software should understand the kind of document it's working with and what's relevant. But the **traditional OCR software can struggle to scale document classification efforts**. Hence software bots should be trained with cognitive abilities by **leveraging machine learning and deep learning techniques to make the OCRs more intelligent**. **ML-based OCR solutions can identify a document type and match it against a known document type used by your business**. They can also parse and understand blocks of text in unstructured documents. Once the solution knows more about the document itself, it can begin to extract relevant information based on intent and meaning.
5. **Better Data Extraction and Classification:** **Data extraction is the core of Document Understanding**. As discussed in the previous section on Integrating RPA's with OCR in this step, opt the data extraction technique based on the type of document. **Through RPAs, we can easily configure which extractor to use, whether a rule-based or ML-based or a hybrid model OCR technique**. **Based on the confidence and**

performance metrics that are returned after the information extraction, the software robots will save them in our desired format for further analysis. Below is an image of how we can configure extractors and set confidence level in an RPA tool by UiPath.

Configure Extractors

Configure which extractors you want to apply to each document type and field.
To activate extractors for certain fields, check the appropriate boxes in the configurator.
For extractors that use your defined taxonomy for configuration and data extraction, entering the unique IDs for document types and fields is optional.
For extractors that have their own internal taxonomy, provide the internal taxonomy unique IDs for both document types and fields for which the extractor will be activated.

Document Types and Fields	Regex Based Extractor Minimum Confidence % 90	Form Extractor Minimum Confidence % 90	Intelligent Form Extractor Minimum Confidence % 0	Invoices ML Extractor Minimum Confidence % 90	Receipts ML Extractor Minimum Confidence % 0	Purchase Orders ML Extractor Minimum Confidence % 0
► Purchase Orders	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> N/A	<input type="checkbox"/> N/A	<input checked="" type="checkbox"/> N/A
► Receipts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> N/A	<input checked="" type="checkbox"/> N/A	<input type="checkbox"/> N/A
▼ Invoices	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> N/A	<input type="checkbox"/> N/A	<input type="checkbox"/> N/A
Date	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> date	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Name	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> name	<input type="checkbox"/>	<input type="checkbox"/>
Invoice No.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> invoice-no	<input type="checkbox"/>	<input type="checkbox"/>
PO No.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> po-no	<input type="checkbox"/>	<input type="checkbox"/>
Billing Address	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> billing-addr	<input type="checkbox"/>	<input type="checkbox"/>
Shipping Address	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> shipping-addr	<input type="checkbox"/>	<input type="checkbox"/>
Vendor Address	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> vendor-addr	<input type="checkbox"/>	<input type="checkbox"/>
Currency	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> currency	<input type="checkbox"/>	<input type="checkbox"/>
Tax Amount	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> tax	<input type="checkbox"/>	<input type="checkbox"/>
Total	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> total	<input type="checkbox"/>	<input type="checkbox"/>

Save Cancel

Choosing necessary Extractors for Document Understanding in UI Path

6. Validation and Empowering Insights: OCR and Machine Learning models are not a hundred percent accurate in terms of information extraction, hence adding a layer of human intervention with the help of robots can solve the problem. The way this validation works is that whenever the robots deal with low accuracy and exceptions, it immediately raises a notification to the action center where an employee can receive a request to validate data or handle exceptions and can solve any uncertainties in a matter of clicks. **Further, we can unlock the potential of Artificial Intelligence to document data over time to make predictions, and identify potential anomalies that may indicate fraud, duplication, and other errors.**

Benefits of integrating robots with Document Understanding

1. **Automate Process:** The key reason for integrating bots for document understanding is to automate the entire process from start to end. All we need to do is create a workflow for the bots to learn, sit back, and relax. During the validation process, we might need to address the issues that are notified by the bots where any errors or frauds are identified.
2. **Bots with Machine Learning:** During the automation process, we can make the bots resilient to machine learning. Meaning the robots can also learn how Machine Learning models are performing and thereby enhance the models to achieve higher accuracy and performance for text and information extraction of documents.
3. **Process Wide Range of Document Processing:** For general tasks like table and information extraction, we'll have to create different deep learning pipelines for different types of documents. This leads to building multiple applications and deploying various models on different servers, which requires a lot of effort and time. When the bots are in the picture for a wide range of documents, we could only have a single pipeline wherein the bots can classify them and then use the appropriate model for different tasks. We can also integrate various services through APIs and communicate with other organizations in terms of fetching the data.
4. **Easy to Deploy:** For document understanding after the pipelines are created, the deployment process is just a minute. We can either have APIs exported by bots after training, or else we can have a build a custom RPA solution that can be used in our local systems. This type of deployment can also optimize the enterprises and can reduce the expenditure with very minimal risks.

Enter Nanonets

NanoNets is a Machine Learning platform that allows users to capture data from invoices, receipts, and other documents without

any template setup. We have state of the art deep learning and computer vision algorithms running at the back that can handle any kind of document understanding tasks like **OCR, table extraction, key-value pair extraction**. They are usually exported as APIs or can be deployed on-premises based on different use cases. Here are a few examples,

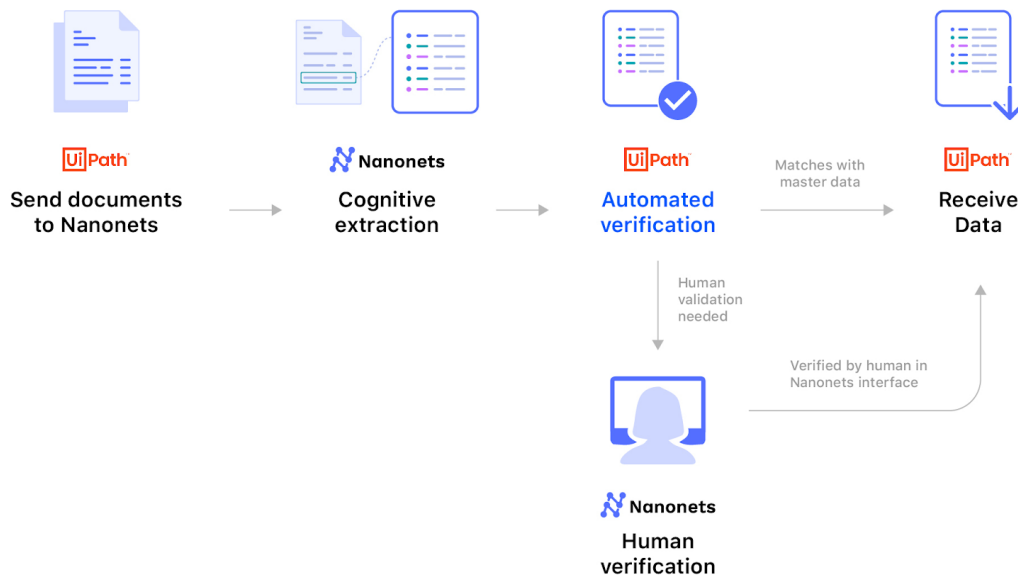
1. Invoice Model: Identify key fields from Invoices like Buyers Name, Invoice Id, Date, Amount etc.
2. Receipts Model: Identify key fields from Receipts like Sellers Name, Number, Date, Amount etc.
3. Driving License(USA): Identify key fields like License No, DOB, Expiry Date, Issue Date etc.
4. Resumes: Extract experience, education, skill sets, candidate info etc.

To make these workflows faster and robust, we use UiPath, an RPA tool for seamless automation of your documents without any template. In the next section, we'll go through how you can use UiPath Connect with Nanonets for document understanding. The 3 biggest players in the RPA market are UiPath, Automation Anywhere and Blue Prism. This blog focuses on Uipath.

NanoNets with UiPath

We've learned to create a document understanding pipeline in our previous sections. It requires basic knowledge of OCR, RPA's, and Machine learning, as there are different approaches and algorithms for different tasks at various points. Also, we've to spend much effort building Neural Networks that understand our templates, training, and deploying them. Hence, to be comfortable and automate everything right from uploading documents, classifying them, building OCR, integrating ML models, we at

Nanonets are working on Ui Path to create a seamless pipeline for Document Understanding. Below is an image of how this works.



Now let's review each of these and learn how we can integrate Nanonets with UiPath.

Step 1: Signup at UiPath and Download UiPath Studio

To create a workflow, first, we'll have to create an account in UiPath. If you're an existing user, you can directly log in to your account, redirecting your UiPath dashboard. Next, you'll have to download and install the UiPath Studio (Community Edition), which is free.

Step 2: Download Nanonets Component

Next, to set up your invoice processing pipeline, you'll have to download the Nanonets Connector from the link below.

-> NanoNets OCR - RPA Component

Below is a screenshot of the UiPath Marketplace, and Nanonets Component. Also, to download this, make sure you logged in to UiPath from a Windows operating system.

The screenshot shows the UiPath Connect Marketplace page for the 'NanoNets OCR' Workflow Template. The page layout includes a header with 'UiPath Connect' and navigation links for 'Community', 'Academy', and 'Marketplace'. A search bar and a user profile icon are on the right. The main content area has a breadcrumb trail: 'Connect Home > Marketplace > Components > NanoNets OCR'. The template card for 'NanoNets OCR' features a 'Workflow Template' icon, a star rating, and a 'Download' button. To the right of the card, there is a 'Details' section with tabs for 'Description', 'Media (2)', 'Reviews', and 'Questions'. The 'Description' tab is active, showing a summary and a detailed description of the NanoNets platform. A 'Producers' section lists Shikhar Khanna and Nano Net Technologies Inc. A 'Benefits' section lists four points: automating data capture, easy integration, increasing automation to 90-95%, and reducing workload. At the bottom of the details section, there is an email contact for Shikhar Khanna.

UiPath Connect
Community Academy Marketplace

Connect Home > Marketplace > Components > NanoNets OCR

NanoNets OCR
Workflow Template

Description Media (2) Reviews Questions

Summary: Integrate the NanoNets component into your UiPath workflow to capture data from invoices. No template setup! No Keyword based rules! Pure Machine Learning on RPA.

NanoNets is a Machine Learning platform that allows users to capture data from [invoices](#), receipts and [other documents](#) without any template setup. The UiPath workflow is an easy to consume format for our RPA customers to use NanoNets with their bots.

To setup your invoice processing tool all you need is a Free [NanoNets account](#). Get more information on How to get started here on [Setup NanoNets on UiPath](#).

E-Mail me on shikhar (at) nanonets (dot) com for any questions

Producers

Shikhar Khanna
Business Development at [Nano Net Technologies Inc.](#)

Nano Net Technologies Inc.

Benefits

1. Automate data capture from invoices. No template setup required.
2. Integrate easily within your workflows. Make your bots smarter.
3. Increase Automation using AI to 90-95%
4. Reduce workload for BackOffice and AP teams

Download

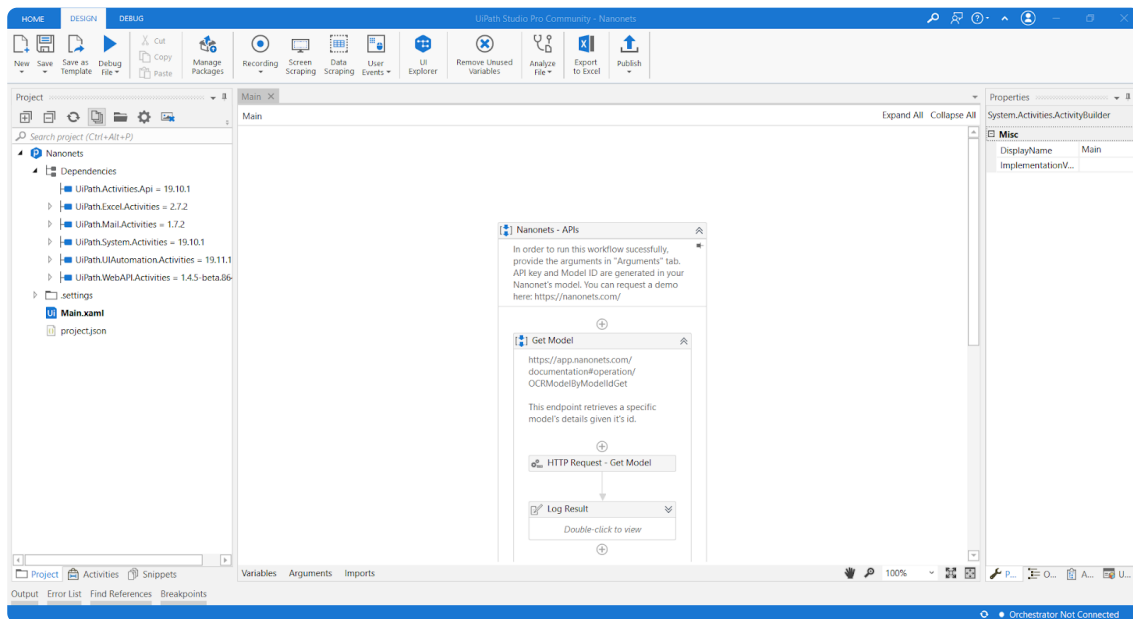
Bookmark

Your downloaded files should contain the files listed below,

```
UiPath OCR Predict
├── Main.xaml
└── project.json
```

Step 3: Open the Main.xaml file Nanonets Component

To check if the Nanonets UiPath is working or not, you can open your Main.xml file from the downloaded Nanonets component using the Ui Path Studio. Then you can see your pipeline already created for you for document processing.



Step 4: Gather your Model ID, API Key and API Endpoint from Nanonets APP

Next, you can use any of the trained OCR models from Nanonets APP and gather the Model ID, API Key, and the endpoint. Below are more details for you to find them quickly.

Model ID: Login to your Nanonets account and navigate to “My Models.” You can train a new model or copy the Application ID of an existing model.

TYPE	Labels	CREATED	ACCURACY	SUMMARY
ocr	buyer_address, buyer	2020-07-09	90.25555%	Model hosted

Application ID: **aec093c4-4840-4c27-8940-d3add200779e** **Model ID**

Images annotated per I... buyer_address: 0, buyer_name: 0, invoice_amount: 0, invoice_date: 0, invoice_num... [Show all](#)

API Calls Processed: 0

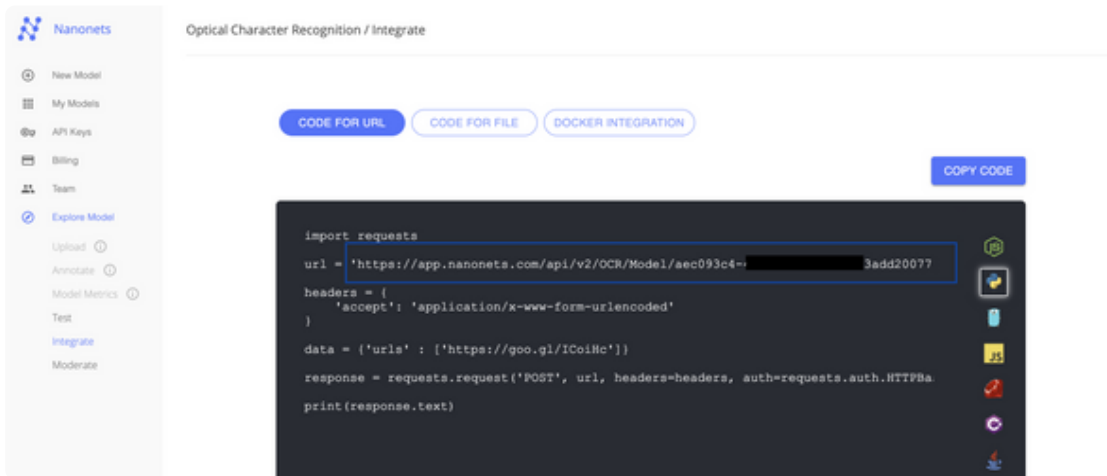
API Calls Left: 100

Plan: \$0 for 100 Pages

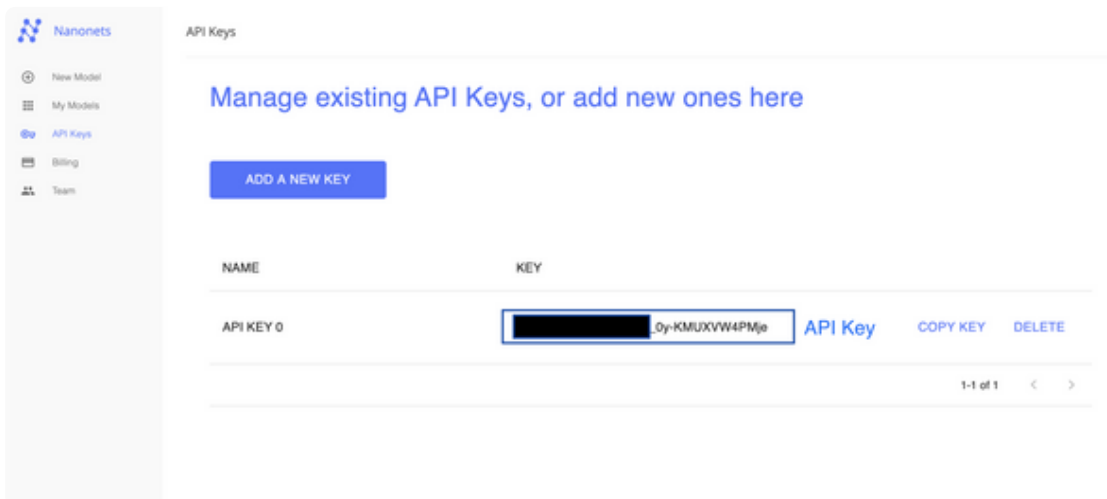
[GO TO MODEL](#) [DELETE MODEL](#) [CHANGE PLAN](#)

API Endpoint: You can choose any existing model and click on Integrate to find your API endpoint. Below is an example of how your endpoints look like.

`https://app.nanonets.com/api/v2/OCR/Model/XXXXXXXXX-4840-4c27-8940-d3add200779e/LabelUrls/`

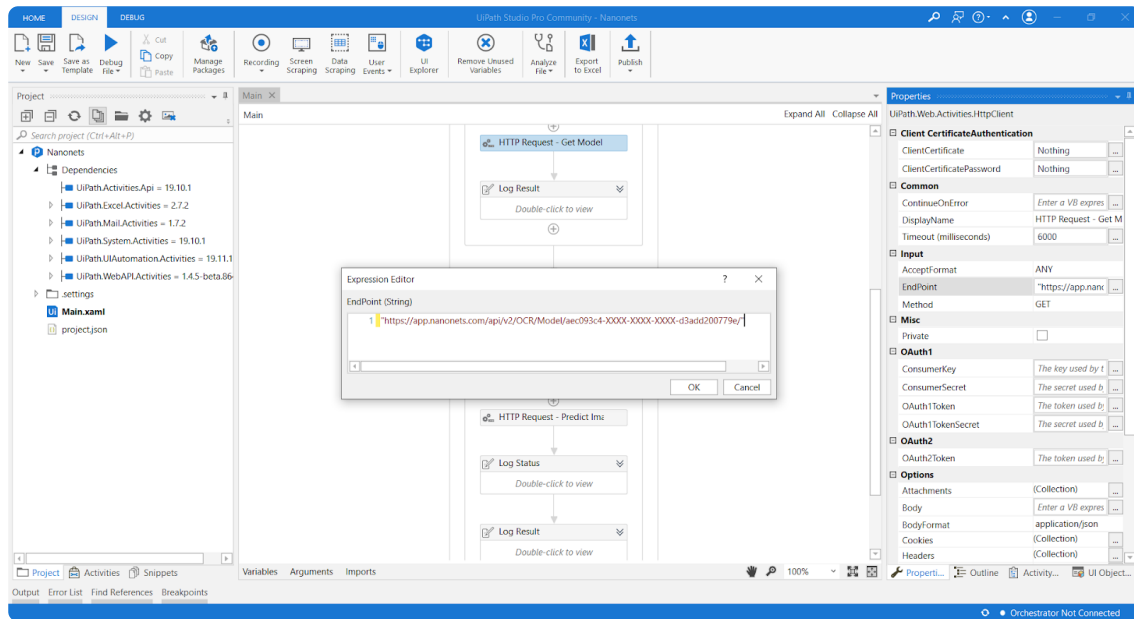


3. API Key: Navigate to the API Key tab, and you can copy any existing API Key or create a new one.

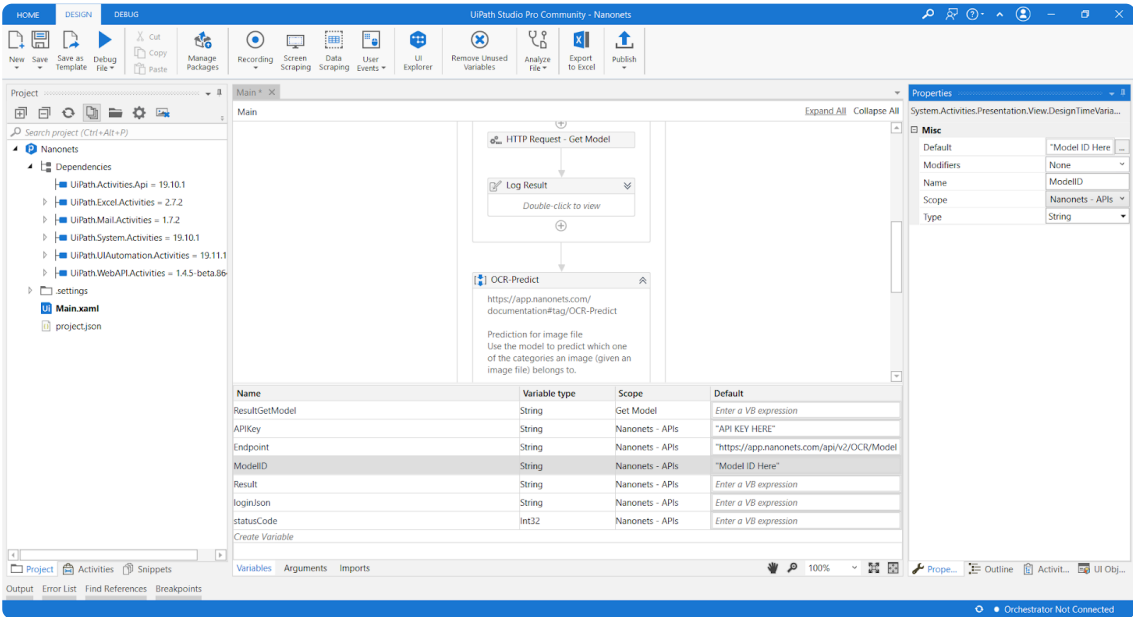


Step 5: Add HTTP Request to get your method and Variables to UI Path

Now to integrate your Model from Nanonets to the UI Path, you'll have the first click on HTTP Request and add the EndPoint, which can be found at left navigation under the Input section. Below is a screenshot.

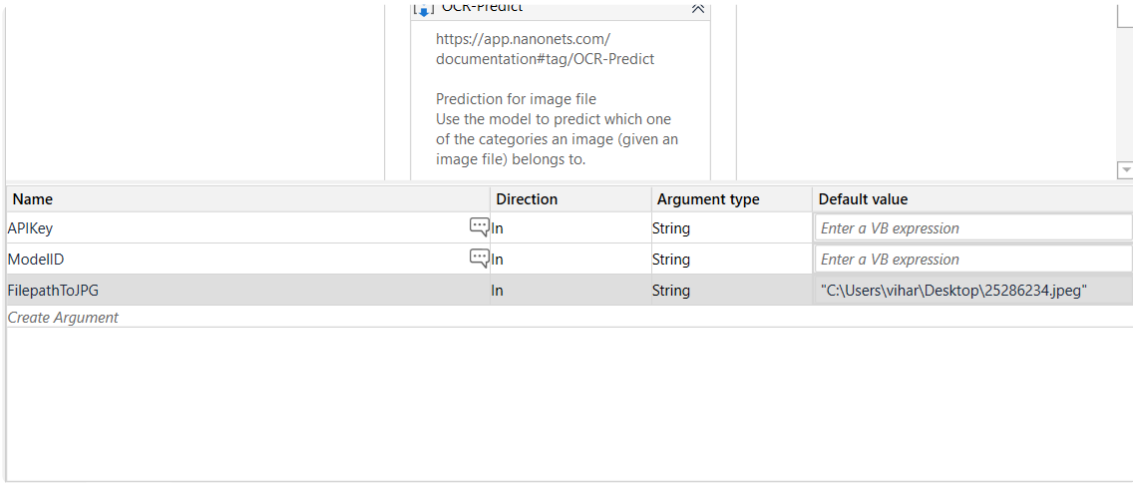


Later, add all your variables to establish a connection from your UiPath studio to the Nanonets API. You can find this section at the bottom pane at the “Variables Tab.” Below is the screenshot, you'll have to update/copy your API Key, End Point and the Model-ID of your model here.



Step 6: Add File Location for Predictions

Lastly, you can add your file location under the attributes tab, as shown in the below screenshot, and hit the play button on your top navigation to predict your outputs.



Voila! Here are our outputs for the document we requested in the below screenshot. To process more, you can simply add your file locations and hit the run button.