Preview Code Blame

<table>
<tr><th>lab</th></tr>
<tr><td><table><tr><th>title</th></tr><tr><td>Explore optical character recognition</td></tr></table></td></tr>
</table>

# Explore optical character recognition

> ⓘ Note
>
> To complete this lab, you will need an Azure subscription in which you have administrative access.

A common computer vision challenge is to detect and interpret text in an image. This kind of processing is often referred to as *optical character recognition* (OCR). Microsoft's Read API provides access to OCR capabilities.

To test the capabilities of the Read API, we'll use a simple command-line application that runs in the Cloud Shell. The same principles and functionality apply in real-world solutions, such as web sites or phone apps.

## Use the Computer Vision Service to Read Text in an Image

The **Computer Vision** cognitive service provides support for OCR tasks, including:

- A **Read** API that is optimized for larger documents. This API is used asynchronously, and can be used for both printed and handwritten text.

## Create a *Cognitive Services* resource

You can use the Computer Vision service by creating either a **Computer Vision** resource or a **Cognitive Services** resource.

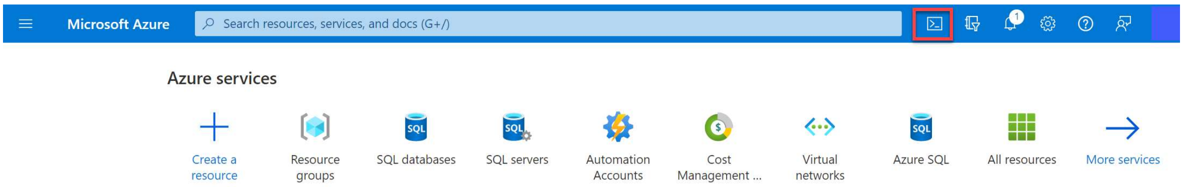
If you haven't already done so, create a **Cognitive Services** resource in your Azure subscription.

1. In another browser tab, open the Azure portal at https://portal.azure.com, signing in with your Microsoft account.
2. Click the **＋Create a resource** button, search for *Cognitive Services*, and create a **Cognitive Services** resource with the following settings:
    - **Subscription**: *Your Azure subscription*.
    - **Resource group**: *Select or create a resource group with a unique name*.
    - **Region**: *Choose any available region*.
    - **Name**: *Enter a unique name*.
    - **Pricing tier**: Standard S0
    - **By checking this box I acknowledge that I have read and understood all the terms below**: Selected.
3. Review and create the resource, and wait for deployment to complete. Then go to the deployed resource.
4. View the **Keys and Endpoint** page for your Cognitive Services resource. You will need the endpoint and keys to connect from client applications.

## Run Cloud Shell

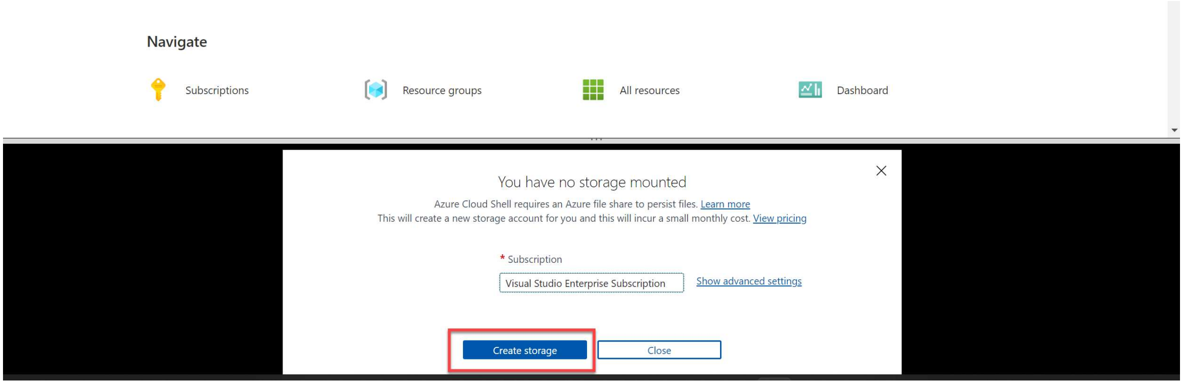
To test the capabilities of the Custom Vision service, we'll use a simple command-line application that runs in the Cloud Shell on Azure.

1. In the Azure portal, select the **[>_]** (*Cloud Shell*) button at the top of the page to the right of the search box. This opens a Cloud Shell pane at the bottom of the portal.



2. The first time you open the Cloud Shell, you may be prompted to choose the type of shell you want to use (*Bash* or *PowerShell*). Select **PowerShell**. If you do not see this option, skip the step.

3. If you are prompted to create storage for your Cloud Shell, ensure your subscription is specified and select **Create storage**. Then wait a minute or so for the storage to be created.

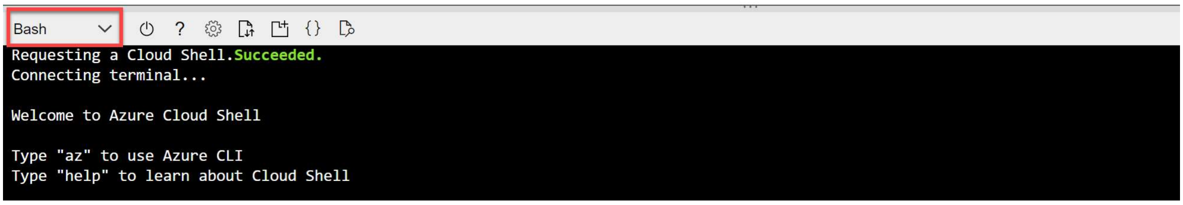


You have no storage mounted

Azure Cloud Shell requires an Azure file share to persist files. Learn more
This will create a new storage account for you and this will incur a small monthly cost. View pricing

* Subscription
Visual Studio Enterprise Subscription

4. Make sure the the type of shell indicated on the top left of the Cloud Shell pane is switched to *PowerShell*. If it is *Bash*, switch to *PowerShell* by using the drop-down menu.

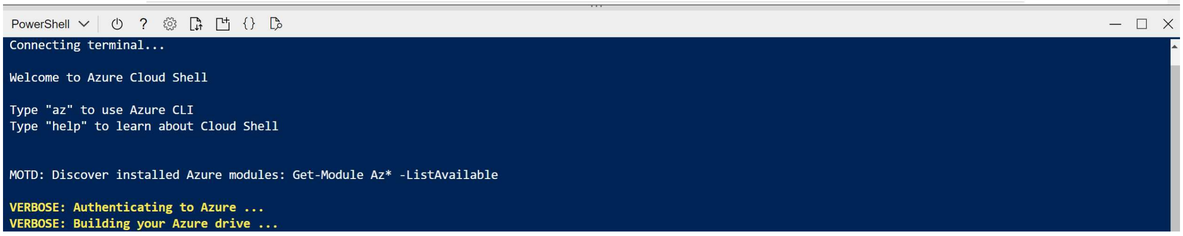


```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell
```

5. Wait for PowerShell to start. You should see the following screen in the Azure portal:



```
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell


MOTD: Discover installed Azure modules: Get-Module Az* -ListAvailable

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
```

## Configure and run a client application

Now that you have a custom model, you can run a simple client application that uses the OCR service.

1. In the command shell, enter the following command to download the sample application and save it to a folder called ai-900.

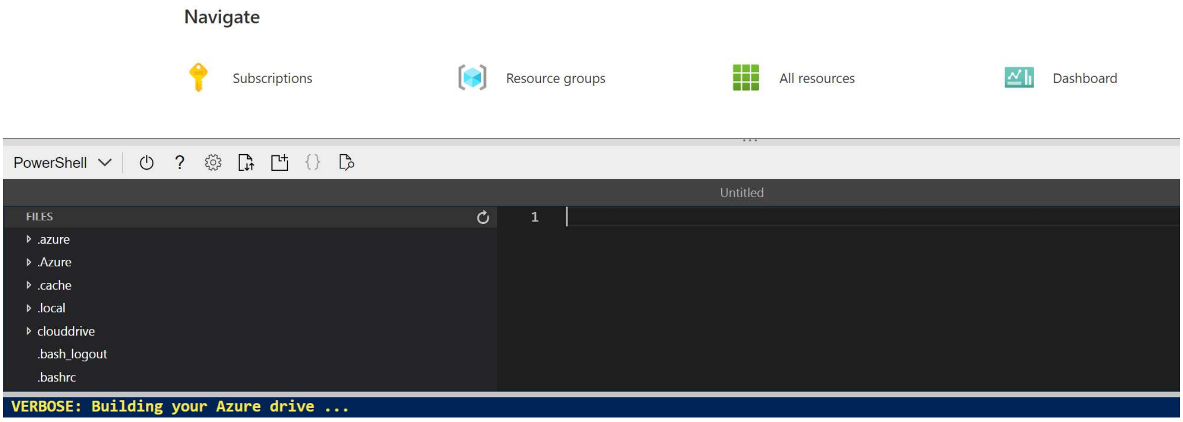
```
git clone https://github.com/Ragowo/SourceCodeAI ai-900
```

> **Tip** If you already used this command in another lab to clone the *ai-900* repository, you can skip this step.

2. The files are downloaded to a folder named **ai-900**. Now we want to see all of the files in your Cloud Shell storage and work with them. Type the following command into the shell:

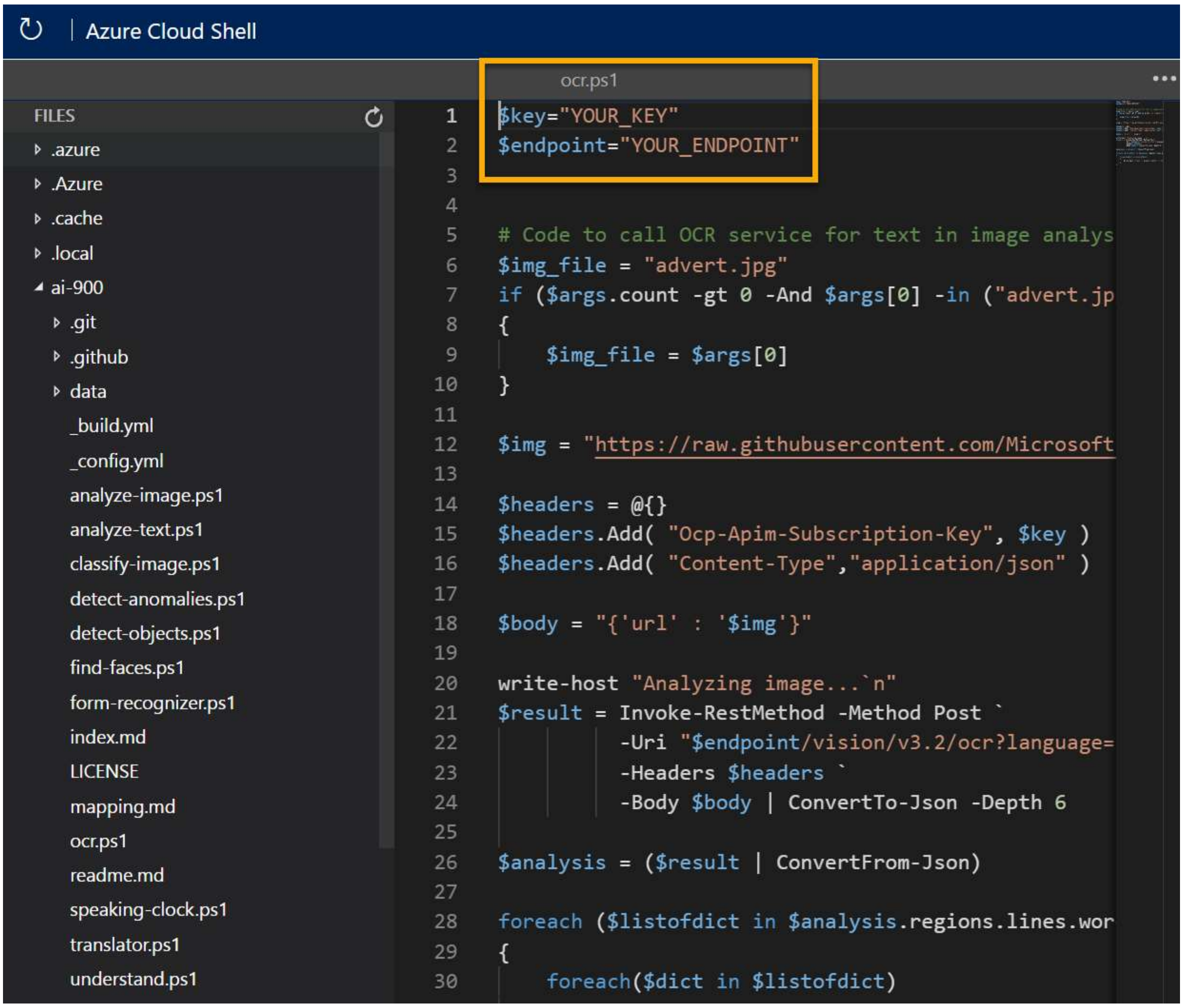
```
code .
```

Notice how this opens up an editor like the one in the image below:



```
VERBOSE: Building your Azure drive ...
```

3. In the **Files** pane on the left, expand **ai-900** and select **ocr.ps1**. This file contains some code that uses the Computer Vision service to detect and analyze text in an image, as shown here:



```
$key="YOUR_KEY"
$endpoint="YOUR_ENDPOINT"


# Code to call OCR service for text in image analys
$img_file = "advert.jpg"
if ($args.count -gt 0 -And $args[0] -in ("advert.jp
{
    $img_file = $args[0]
}

$img = "https://raw.githubusercontent.com/Microsoft

$headers = @{}
$headers.Add( "Ocp-Apim-Subscription-Key", $key )
$headers.Add( "Content-Type","application/json" )

$body = "{'url' : '$img'}"

write-host "Analyzing image...`n"
$result = Invoke-RestMethod -Method Post `
          -Uri "$endpoint/vision/v3.2/ocr?language=
          -Headers $headers `
          -Body $body | ConvertTo-Json -Depth 6

$analysis = ($result | ConvertFrom-Json)

foreach ($listofdict in $analysis.regions.lines.wor
{
    foreach($dict in $listofdict)
```

4. Don't worry too much about the details of the code, the important thing is that it needs the endpoint URL and either of the keys for your Cognitive Services resource. Copy these from the **Keys and Endpoints** page for your resource from the Azure portal and paste them into the code editor, replacing the **YOUR_KEY** and **YOUR_ENDPOINT** placeholder values respectively.

   > **Tip** You may need to use the separator bar to adjust the screen area as you work with the **Keys and Endpoint** and **Editor** panes.

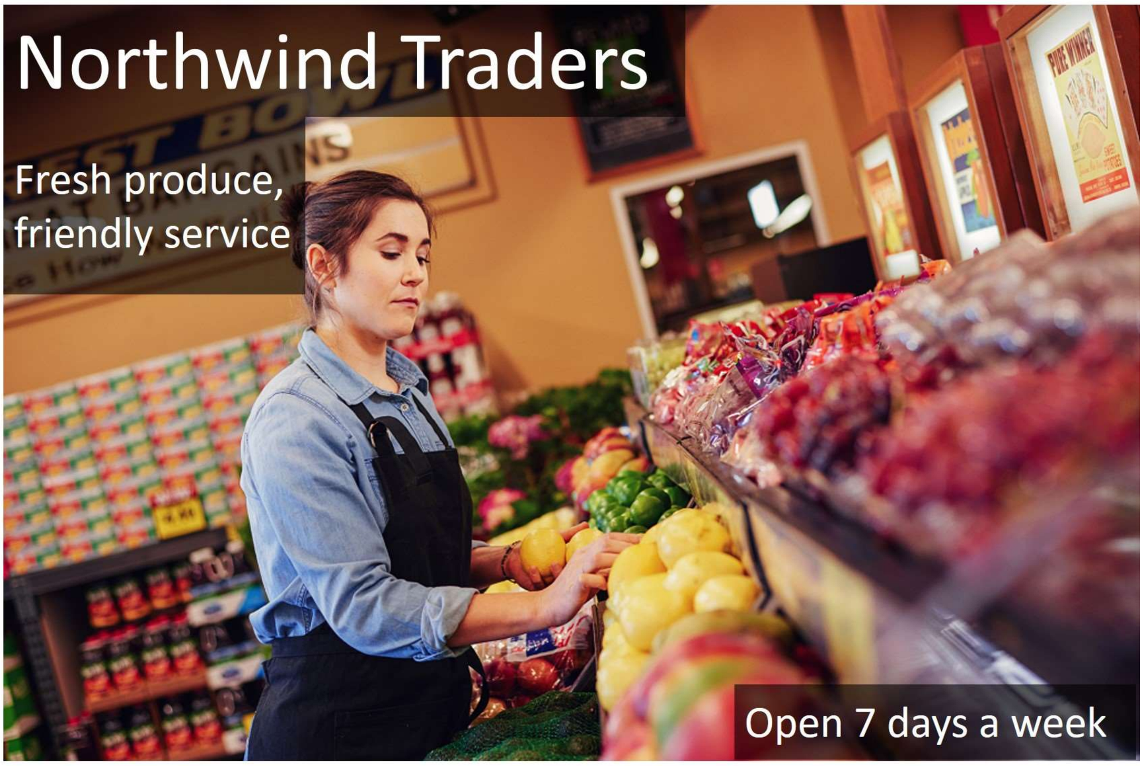
   After pasting the key and endpoint values, the first two lines of code should look similar to this:

   ```
   $key="1a2b3c4d5e6f7g8h9i0j...."
   $endpoint="https..."
   ```

5. At the top right of the editor pane, use the **...** button to open the menu and select **Save** to save your changes. Then open the menu again and select **Close Editor**. Now that you've set up the key and endpoint, you can use your Cognitive Services resource to extract text from an image.

   Let's use the **Read** API. In this case, you have an advertising image for the fictional Northwind Traders retail company that includes some text.

The sample client application will analyze the following image:

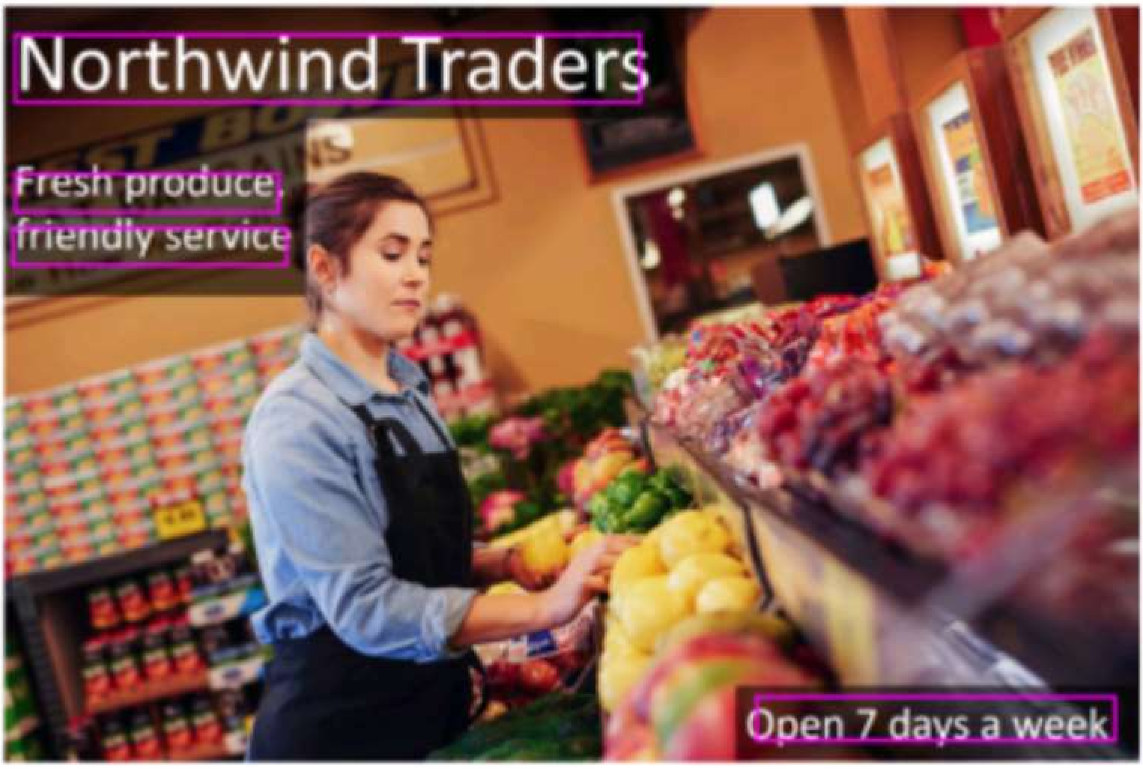


6. In the PowerShell pane, enter the following commands to run the code to read the text:

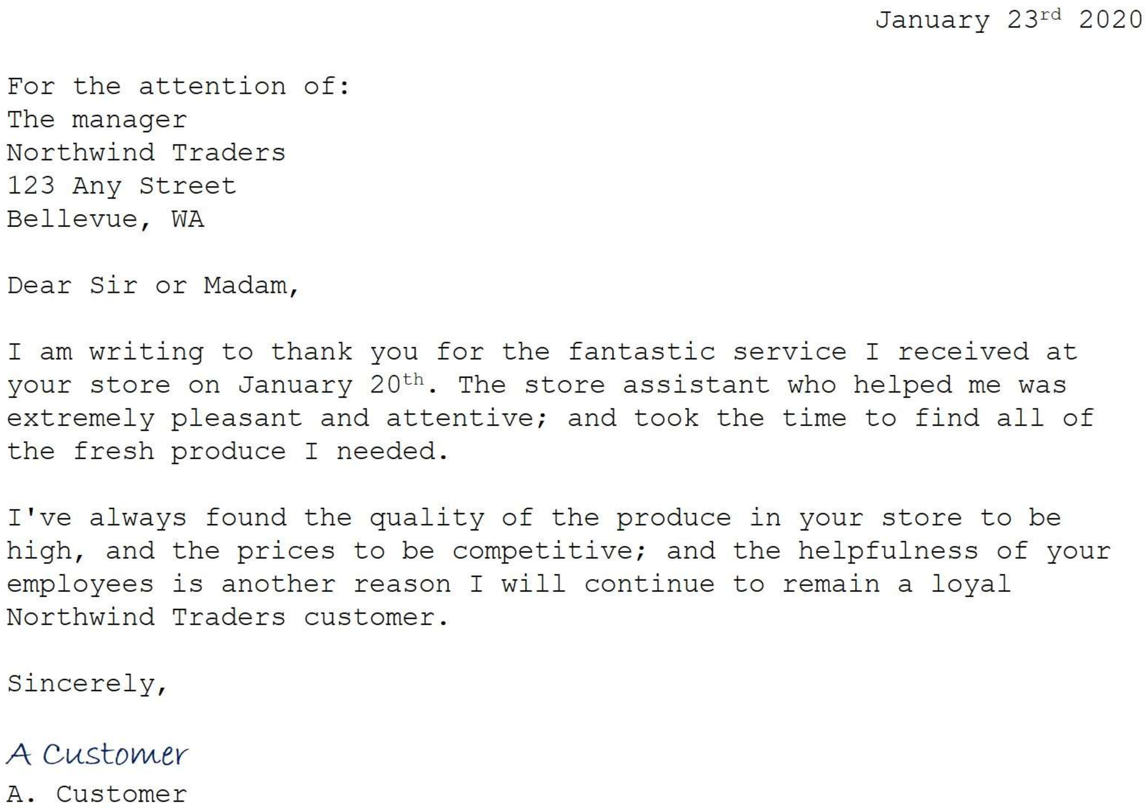
```
cd ai-900
./ocr.ps1 advert.jpg
```

7. Review the details found in the image. The text found in the image is organized into a hierarchical structure of regions, lines, and words, and the code reads these to retrieve the results.

   Note that the location of text is indicated by the top- left coordinates, and the width and height of a *bounding box*, as shown here:



8. Now let's try another image:



January 23rd 2020

For the attention of:
The manager
Northwind Traders
123 Any Street
Bellevue, WA

Dear Sir or Madam,

I am writing to thank you for the fantastic service I received at your store on January 20th. The store assistant who helped me was extremely pleasant and attentive; and took the time to find all of the fresh produce I needed.

I've always found the quality of the produce in your store to be high, and the prices to be competitive; and the helpfulness of your employees is another reason I will continue to remain a loyal Northwind Traders customer.

Sincerely,

A Customer
A. Customer

To analyze the second image, enter the following command:

```
./ocr.ps1 letter.jpg
```

9. Review the results of the analysis for the second image. It should also return the text and bounding boxes of the text.

## Learn more

This simple app shows only some of the OCR capabilities of the Computer Vision service. To learn more about what you can do with this service, see the OCR page.