Niloy E. Siddiquee Khandoker
University of Bradford
12/05/2022

Table of Contents

1. Introduction

Handwriting recognition (HWR)[1] refers to the ability of robots or computers to understand human handwriting entries from physical data files into text. Most of the time, the computer receives input in the form of a picture of the handwriting, which is then processed by software to establish its nature. The goal of handwriting recognition is to automatically read coherent handwritten input.

2. Dataset and research

The chosen dataset for the development of this project is the "NIST Special Database 19". This database features 3600 writer's Handprinted Sample Forms, 810000 character images separated from their forms, ground truth classifications for those images and reference forms for additional data gathering.

This database was found in one biggest website related to machine learning, Kaggle. This page allows developers to access and share datasets, study, and construct models in a web-based machine learning environment, collaborate with some other data scientists and machine learning experts, and enter data science competitions. Link for the dataset: https://www.kaggle.com/datasets/sachinpatel21/az-handwritten-alphabets-in-csv-format

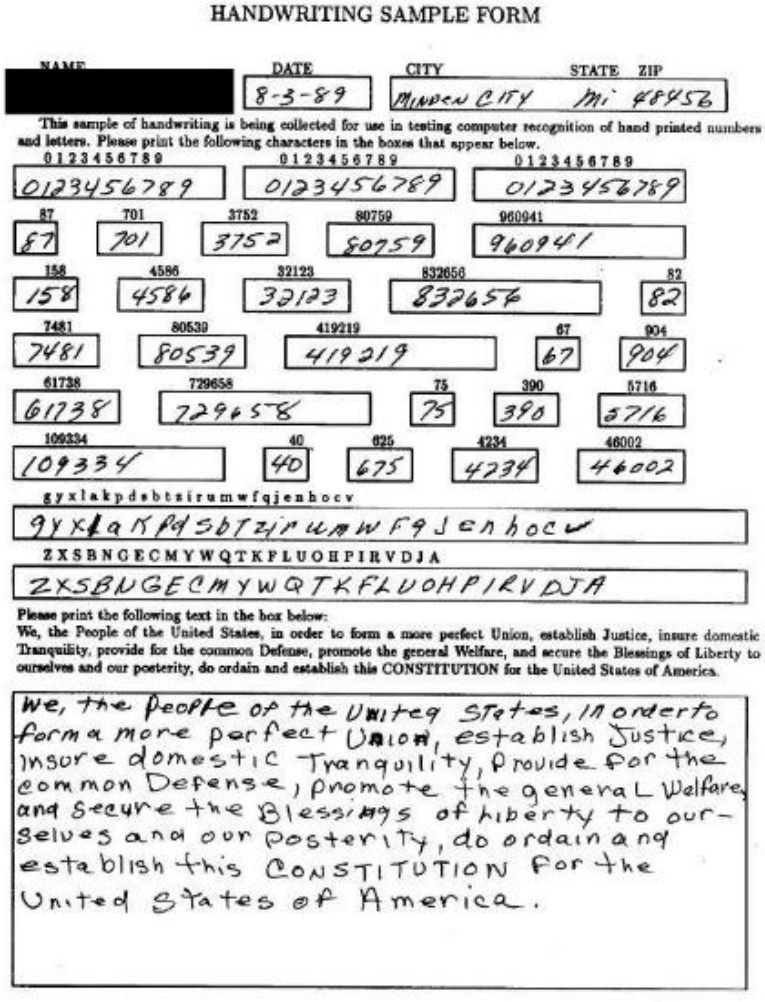

Figure 1: Handwriting sample form (nist.gov)

This photo shows the handwriting sample forms which was given to every writter to fill in order to complete the dataset. As shown, each of them was obligated to write down the random numbers and random letters followed by a full paragraph.

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	...	0.639	0.640	0.641	0.642	0.643	0.644	0.645	0.646	0.647	0.648
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
372445	25	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
372446	25	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
372447	25	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
372448	25	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
372449	25	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

372450 rows × 785 columns

Figure 2: NIST dataset

This figure shows the dataset in csv format. As we can see, it is formed by 372450 rows and 785 collumns. It actually contains handwritten images with the size of 28x28 pixels. It is divided in 26 folders with letters from A to Z.

2.1 Previous work

Many researchers have been done using the NIST dataset since it allows users to program and analyze a reliable set of data. One of the most important work in this area was made by Patrick J Grother in 1995. While being student of the National Institute of Standards and Technology, he studied and investigated the dataset profoundly. He described in detail the data hierarchies by field type, by hexadecimal class and by merged classes among others.

The fields of each form were segmented into isolated characters where each image was automatically assigned a classification. He stated that all the referenced character images was manually checked twice by different people since image's class was being changed upon detection of an error (1995)[2].

2.2 Questions

As I described in the introduction, the most common use of HWR nowadays is the capacity of smart phones to perceive the input from the user directly on the touch screen. This makes our life easier when we have to write down notes or texts in our Notes Application. Many famous universities are in favor of using less paper for future generations. This is a clear sign that HWR is going to be really important when a teacher or module coordinator speaks in class and people has to take notes. It is way quicker to handwrite on your phone or tablet, which can be transformed to word or text file.

Each person has a different handwriting. For example, if a person writes the letter C in a closed way it could get confused by the letter G. This could cause systems to not properly detect the entry. So, is it possible to

detect handwriting after training and evaluating the dataset? My case study consists of analyzing the NIST Special Dataset to train a model that can properly identify a letter.

3. Data Analysis

In this section we will study the variables and entries inside our dataset.

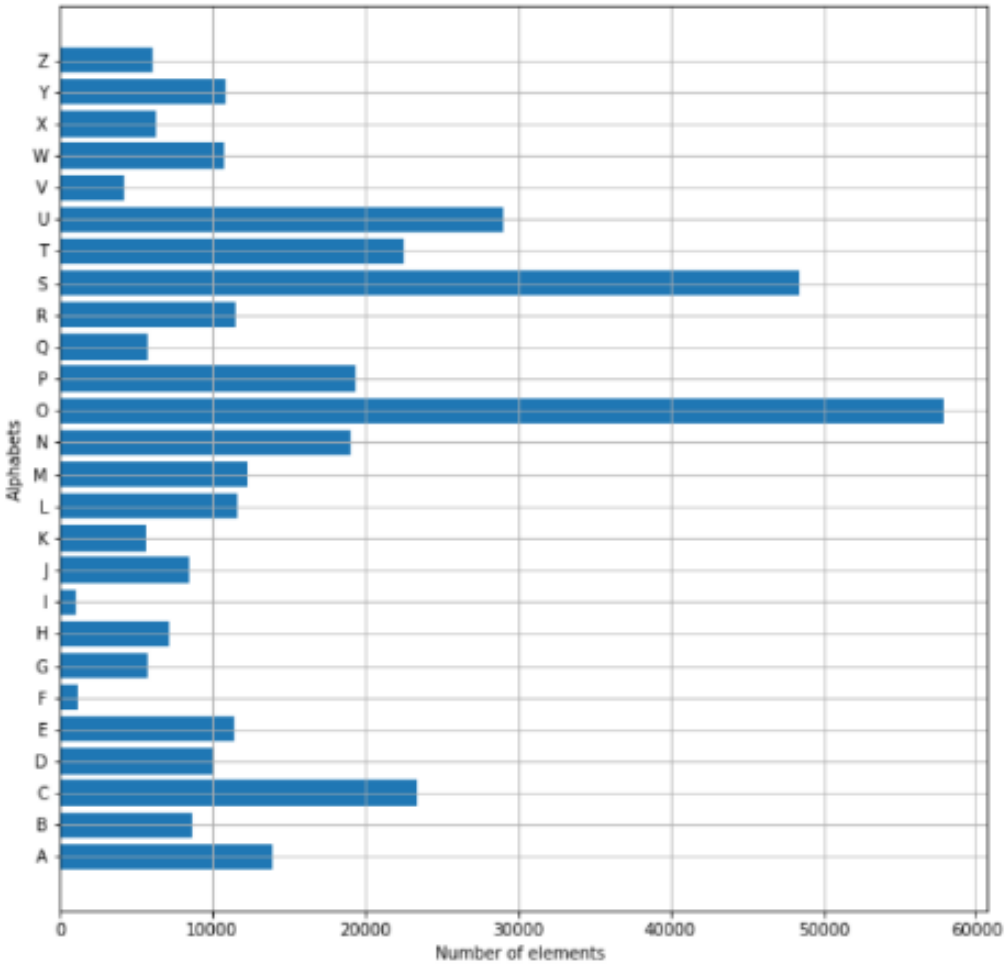

Figure 3: Plotting alphabet letters

A plot is a graphical representation of a data collection, usually in the form of a graph demonstrating the relationship between two or more factors. The plot can be created by hand or by using a computer. Plotter machines, both mechanical and electronic, were utilized in the past. Graphs are a pictorial representation of the correlation that are particularly important for humans because they allow them to quickly gain an understanding that would otherwise be impossible to derive from lists of data. Wrist (2014)[3] stated that it is really important to select the correct graph type based on the kind of data to be presented.

Figure number 3 presents the plotting of the number of alphabets in the dataset. As shown above, we can diferenciate the number of times a letter is repeted in a horizontal bar plot. It demonstrates that some letters are used more often than others. We can appreciate some letters that are used frequently, such as “C”, “T”, “U” or “S”, but without a doubt, there is a clear reiteration of the letter “O”. On the other hand, we can see the absence of letters like “I” or “F”.

This graph also correlates with an analysis of the Concise Oxford Dictionary (9th Edition, 1995)[4] where the letter "O" occupies the 5th place considering its frecuency in English words.

	labels	1	2	3	4	5	6	7	8	9
count	372450.000000	372450.0	372450.0	372450.0	372450.0	372450.0	372450.0	372450.0	372450.0	372450.0
mean	13.523490	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
std	6.740824	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
min	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25%	10.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50%	14.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
75%	18.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
max	25.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

8 rows × 785 columns

Figure 4: Data description

Figure number 4 studies relevant values for data description.

The dataset's maximum and minimum values are both relatively simple metrics. The difference between the maximum and minimum value is called the data range. The minimum value of our dataset is 0 while the maximum is 25.

Mean (also called average) sums up all the values in your column and divides them by the number of values. In our case the mean is 13,52.

The standard deviation is a measurement of a set of values' variation or dispersion. It measures how spread out the data is. The standart deviation (std) is 6,7.

```
data = pd.read_csv("/content/gdrive/MyDrive/A_ZHandwrittenData.csv")
data.info(verbose=False, show_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 372450 entries, 0 to 372449
Columns: 785 entries, 0 to 0.648
dtypes: int64(785)
memory usage: 2.2 GB
```

Figure 5: null values

Using verbose we can identify that our data set is a clean data set since it does not contain any null value.

4. Practical work and results

Two main technologies have been used during the development of this project: Python and IDE Google Research (Google Colab).

Python has recently become one of the most widely used programming languages on the planet. Everything from machine learning to website development and software testing uses it. It is suitable for both developers and non-developers. Python is versatile, since it is a general-purpose language so it can be used

to produce wide range of programs and is not focused on any particular task. RedMonk is a prestigious forum for people related to coding and web development. O'Grady wrote an article in 2021 conducting a study about the different coding languages nowadays. Python end up in the second place just after Javascript (2021)[5]. One of the most important aspect of Python is that is a friendly language for machine learning, data science and statistical calculations purposes

Google Colab is a web-based Python editor that allows anyone to write and run arbitrary Python code. It's notably useful for machine learning, data analysis, and education.

```
import matplotlib.pyplot as plt
import cv2
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Flatten, Conv2D, MaxPool2D, Dropout
from keras.optimizers import *
from keras.callbacks import ReduceLROnPlateau, EarlyStopping
from tensorflow.keras.utils import to_categorical
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle
```

Figure 6: required frameworks

For the development of the project we need to import the required frameworks:

- Numpy: used to conduct a wide range of array-based mathematical operations. It provides a vast library of high-level mathematical functions that work on these arrays and matrices, as well as powerful data structures that guarantee efficient calculations with arrays and matrices (2022)[6].

- Cv2: used for face identification and detection, licence plate reading, photo editing, advanced robotic vision, optical character recognition, and many other applications.

- Keras: provides uniform and straightforward APIs, reduces the number of user steps for common use cases, and delivers clear and responsive error messages. It comes with a lot of documentation and developer instructions. It is really easy to learn and use. A survey made in 2019 to the top teams in Kaggle to rank frameworks for machine learning and deep learning purposes showed that Keras ranked the first (2022)[7].

- Tensorflow: provides a set of Python procedures for developing and training models. Loads and preprocesses data. It was build originally for the development of numerical computations.

- Matplotlib: used mainly for the creation of visual representations, graphics, charts and plots in using python. One key function of Matplotlib is that it can make 2D plots from entries in arrays.

- Pandas: is a rapid and effective dataframe for data handling with integrated indexing. One main function is that it allows to read CSV or text files, reshape data, merge and join data sets.

```
X = data.drop('0',axis = 1)
y = data['0']
```

Figure 7: data split

We devide our data into images and labels. We are going to drop the ‘0’ because it contains the labels and use it in the y form the labels.

```
[ ] train_x, test_x, train_y, test_y = train_test_split(X, y, test_size = 0.2)
    train_x = np.reshape(train_x.values, (train_x.shape[0], 28,28))
    test_x = np.reshape(test_x.values, (test_x.shape[0], 28,28))
```

```
[ ] print("Train data shape: ", train_x.shape)
    print("Test data shape: ", test_x.shape)

    Train data shape:  (297960, 28, 28)
    Test data shape:  (74490, 28, 28)
```

Figure 8: reshape the data

In the above segment we are splitting the data into training and testing data set using train_test_split()

If we want the image data to be displayed as an image we have to reshape the train and test data, which was represented as 784 columns inside the CSV file, we need to transform it to 28x28 pixels.

We have now reshaped our data set, having 297960 training data and 74490 test data, which can be put now in the model.

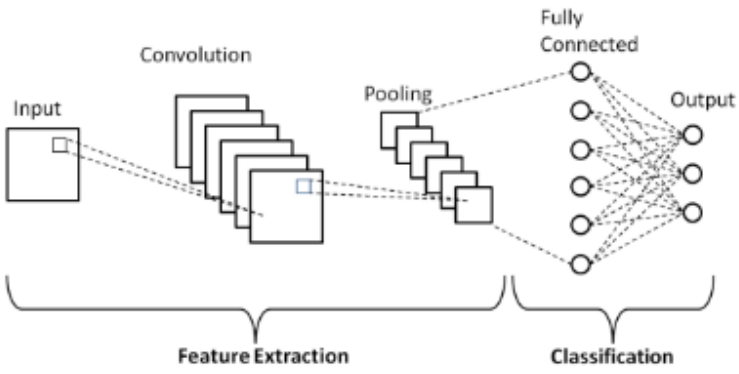

Figure 9: CNN

To complete this project I decided to use a CNN (Convolutional Neural Networks) classification. Compared to other classifications, like ANN (Artificial Neural Networks), CNN is more popular. It takes the raw pixel data from an image, trains the model, and then extracts the features for improved categorization. I focused on the convolution, pooling and fully connected layers of the CNN classification.

Convolutional layer is the pain part of CNN. It is used mainly to decrease the size of the image and to process pixel data and image recognition. It reduces the image withoud losing the relation between pixels.

After each convolution layer, a pooling layer is added to aid in the case of an overfitting problem. Max Pooling is a technique for reducing the size by selecting the maximum. We call it a fully connected network when each parameter is joined to each other.

```
[ ]  train_yOHE = to_categorical(train_y, num_classes = 26, dtype='int')
     print("New shape of train labels: ", train_yOHE.shape)
     test_yOHE = to_categorical(test_y, num_classes = 26, dtype='int')
     print("New shape of test labels: ", test_yOHE.shape)

     New shape of train labels:  (297960, 26)
     New shape of test labels:  (74490, 26)
```

Figure 10: float transformation

We have to convert the float to categorical values because we are using the Convolutional Neural Networks.

```
[ ]  model = Sequential()
     model.add(Conv2D(filters=32, kernel_size=(3, 3), activation='relu', input_shape=(28,28,1)))
     model.add(MaxPool2D(pool_size=(2, 2), strides=2))
     model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding = 'same'))
     model.add(MaxPool2D(pool_size=(2, 2), strides=2))
     model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu', padding = 'valid'))
     model.add(MaxPool2D(pool_size=(2, 2), strides=2))
     model.add(Flatten())
     model.add(Dense(64,activation ="relu"))
     model.add(Dense(128,activation ="relu"))
     model.add(Dense(26,activation ="softmax"))
```

```
[ ]  model.compile(optimizer = tf.keras.optimizers.Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
     history = model.fit(train_X, train_yOHE, epochs=1,  validation_data = (test_X,test_yOHE))

     9312/9312 [==============================] - 409s 40ms/step - loss: 0.1619 - accuracy: 0.9560 - val_loss: 0.0876 - val_accuracy: 0.9750
```

```
[ ]  print("The validation accuracy is :", history.history['val_accuracy'])
     print("The training accuracy is :", history.history['accuracy'])
     print("The validation loss is :", history.history['val_loss'])
     print("The training loss is :", history.history['loss'])

     The validation accuracy is : [0.9749630689620972]
     The training accuracy is : [0.956000804901123]
     The validation loss is : [0.0875934362411499]
     The training loss is : [0.1618584841489792]
```

Figure 11: CNN model

Figure 11 represents the creation of the CNN model. I compliled the model and defined the optimized function and loss. This was done using an optimizer called Adam, which uses the best qualities of AdaGrad and RMSProp algorithms. The results was printed showing us that the validation has an accuracy of 97%, training accuracy of 95%, validation loss of 8% and the training loss of 16%.

After training and testing our models now we are ready to do some prediction, creating 9 subplots to access some data from the data set using model.predict().

```
fig, axes = plt.subplots(3,3, figsize=(8,9))
axes = axes.flatten()
for i,ax in enumerate(axes):
    img = np.reshape(test_X[i], (28,28))
    ax.imshow(img, cmap="Greys")

    pred = word_dict[np.argmax(test_yOHE[i])]
    ax.set_title("Prediction: "+pred)
    ax.grid()
```

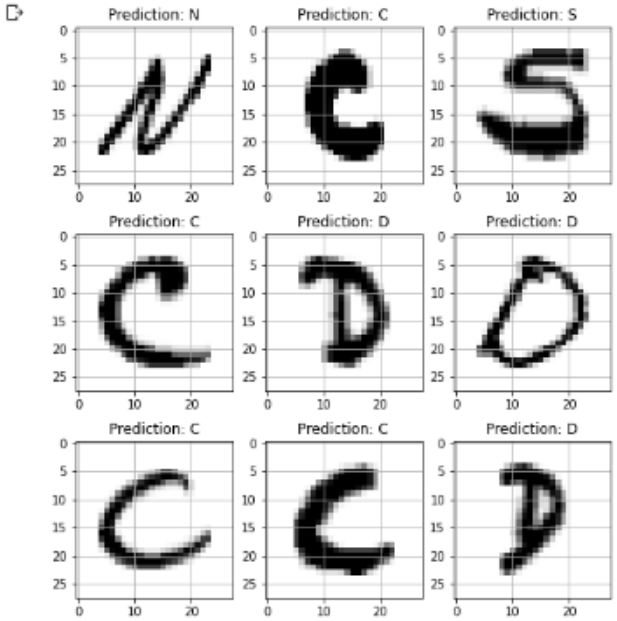

Figure 12: letter prediction

Figure 12 represents the model prediction. Every letter from the 9 subplots was correctly guessed by the model

5. Critical Review

This report has covered the use of the NIST data set to perform a series of machine learning actions to predict letters. Some difficulties were found during the realization of this project. I was receiving many errors and was stuck with the project for some time because I was not getting the correct results changing float values to categorical values.

After correctly guessing 9 subplots from the data set, I realized that it has the potential to predict external images. I tried importing a random image with a letter and reshape it, but I got some issues trying to do the reshaping. This feature can be considered for future work. If we manage to import a random image with one letter and the model can guess the letter correctly, we could use it to predict proper words and eventually proper phrases.

As I mentioned in the introduction, HWR is the capacity of computers to understand human handwriting. Nowadays, HWR is mostly used in smartphones when we write down using the touch screen. Depending on the degree that you are studying, taking written notes are much quicker than writing on a computer because we synthesize the phrases as we want.

Some Universities are in favor of reducing the amount of paper we use since almost everything is digitized. If we imagine a class full of medical students taking complex notes about biology in a mobile phone or tablet, we want to make sure that the notes we take are transformed into a text file correctly predicting every input. That is why a proper training and testing was necessary to study a broad range of letters.

Nowell (2017)[8] already investigated in this area of how could machine learning could gain profit combined with HWR. He stated that in an increasingly connected world of international trade, language barriers are a prevailing problem. At TRG we believe using machine learning techniques such as HWR could be a major step in enhancing the efficiency of supply chain management for the businesses we work with.

6. Bibliography

1. Anonymous (2021) *Handwriting Recognition (HWR).* Technopedia. https://www.techopedia.com/definition/196/handwriting-recognition-hwr#:~:text=Handwriting%20recognition%20is%20the%20ability,then%20interpret%20this%20as%20text Accessed 6 May 2022.

2. Grother, P. J. (1995) NIST Special Database 19. Handprinted Forms and Characters Database. Page 5 https://www.nist.gov/system/files/documents/srd/nistsd19.pdf Accessed 6 May 2022.

3. J. Wrist Surg (2014) *The Effective Use of Graphs.* National Library of Medicine. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4078179/ Accessed 7 May 2022.

4. Anonymous (1995) The frequency of the letters of the alphabet in English. https://www3.nd.edu/~busiforc/handouts/cryptography/letterfrequencies.html Accessed 7 May 2022.

5. O'Grady, S (2021) *The RedMonk Programming Language Rankings: June 2021.* RedMonk. https://redmonk.com/sogrady/2021/08/05/language-rankings-6-21/ Accessed 7 May 2022.

6. Anonymous (2022) *NumPy: the absolute basics for beginners.* NumPy. https://numpy.org/doc/stable/user/absolute_beginners.html Accessed 7 May 2022.

7. Anonymous (2022) *Why choose Keras?* Keras. https://keras.io/why_keras/#:~:text=Keras%20follows%20best%20practices%20for,learn%20and%20easy%20to%20use. Accessed 7 May 2022.

8. Nowell, J. (2017) *How can machine learning technologies benefit the compliance industry?* Track Record Global. https://www.trackrecordglobal.com/how-can-machine-learning-technologies-benefit-the-compliance-industry/ Accessed 7 May 2022.