*Citation for published version:*
Hamouche, E & Loukaides, E 2018, 'Classification and Selection of Sheet Forming Processes with Machine Learning', *International Journal of Computer Integrated Manufacturing*, vol. 31, no. 9, pp. 921-932. https://doi.org/10.1080/0951192X.2018.1429668

*DOI:*
10.1080/0951192X.2018.1429668

*Publication date:*
2018

*Document Version*
Peer reviewed version

Link to publication

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Computer Integrated Manufacturing on 28 Jan 2018, available online: http://www.tandfonline.com/10.1080/0951192X.2018.1429668

# Classification and Selection of Sheet Forming Processes with Machine Learning

Elia Hamouche, Evripides G. Loukaides*

Department of Mechanical Engineering, University of Bath
Claverton Down, Bath, BA2 7AY
* Corresponding author: e.loukaides@bath.ac.uk

**Abstract**
Sheet metal forming is a critical component of modern manufacturing. The procedure for selecting a suitable manufacturing process to achieve the final geometry of a metal part is unstructured and heavily reliant on human expertise. Similarly, classification and design of new metal forming processes has yet to be automated. In this study, a Machine Learning approach was used for the first time to identify the manufacturing process that formed a part solely from the final geometry. Several Neural Network configurations were tested with different geometry representation methods. The best performing classifier employed a deep Convolutional Neural Network and achieved an accuracy of 89%, namely when the geometry was given through a mapping of the Mean and Gaussian curvatures. The high accuracy rate establishes that automated methods can perform this step between design and manufacture, thus eliminating the need for human experts in matching each product to a suitable forming method.

**KEYWORDS**
Process selection; Machine Learning; Convolutional Neural Networks; Sheet metal forming

## 1. Introduction

Selecting a suitable manufacturing process is a critical task in the design phase of new components. This task is usually undertaken by manufacturing engineers who work closely with design engineers. The use of engineering drawings and heavy reliance on human expertise introduces inefficiencies and prevents automated optimization of this stage of operations. Software used to assist this process has been fairly limited due to rule-based implementation. In this project, a Machine Learning (ML) classifier of sheet metal forming processes was used to aid the manufacturing design process and to replace rule-based implementations.

Metal sheet forming is the process of transforming flat metal sheets to desired 3D geometries through mechanical means. In these processes, the sheet is reshaped without adding or removing material, and its mass remains unchanged. The material undergoes plastic strains to achieve the transformation. These often result in more uniform mechanical characteristics of the final workpiece when compared to contemporary processes such as machining and additive processes (Lange 1985). Sheet metal is used heavily in major industries such as the automotive and aerospace sectors. Example parts include car and lorry bodies, building roofs, engine covers, aeroplane fuselages and wings. Various processes are used to form rolled sheet metal using tools such as dies, rollers and jigs. Common processes include: deep drawing, bending, stretch forming and roll forming (Swift and Booker 2003).

In its current state, the selection of a suitable forming process to shape a thin sheet into the desired geometry is fairly manual and unguided (Swift and Booker 2013). Despite the availability of advanced technologies connecting CAD and CAM systems, integration of those technologies is not as yet full. The current manufacturing process selection stage is done through the explanation of design drawings which hinders the bidirectional flow of information between design and manufacturing engineers. Modern CAPP (Computer Aided Process Planning) systems attempt to tackle this problem by using feature recognition and structured information input by users. These CAPP systems are usually overspecialised and quite inflexible (Kumar and Garg 2011). The limitations of CAPP systems mean that manufacturers rely heavily on human expertise. In turn, this causes the selection of suitable manufacturing processes to be susceptible to errors and inefficiencies.

ML is not a new field, however. With recent growth in computational power, many processes in various industries have seen considerable improvement and advancement by employing artificial intelligence solutions. Recent progress in parallel computing and Graphical Processing Units (GPUs) has renewed interest in artificial Neural Networks (NNs) which have been incredibly successful at tackling complex tasks. Examples of successful implementation of ML in disparate fields are ubiquitous. Processes such as facial recognition (Li and Lu 1999), automated trading (Dempster and Leemans 2006) and equipment fault monitoring (Widodo and Yang 2007) have all been significantly automated and enhanced through the application of ML.

Complex classification tasks usually contain non-linear and inaccessible relationships in their data making them extremely difficult to tackle using a rule-based approach. For example, in facial recognition, a rule-based approach would require detailed information about the structure of a face and its features. This reduces its ability to cope with irregular inputs such as blurry images or unusual angles. An ML based approach would automatically acquire the contextual information. This rule-learning process is known as 'training' and requires labelled training data. In applications where ML has been successful, there has been a shift from reliance on human input to the use of automated and systematic algorithms which only require a database of previous samples labelled with the correct classification.

The use of NNs has facilitated the accurate tagging and classification of images. Complex cognitive tasks which were once exclusive to humans such as as facial recognition, object recognition and counting and scene tagging are now commonly completed by computers and have myriad applications. Simonyan and Zisserman (2014) claimed first and second place in the localisation and classification tracks respectively at the ImageNet Challenge 2014. Their Convolutional Neural Network (CNN) implementation achieved image classification accuracies of 93%, with an average confidence margin greater than 80% for the best trials. The submission made by Google (dubbed GoogleNet) also used a CNN and achieved an overall accuracy of 90% (Szegedy et al. 2015).

Other researchers used a CNN implementation for handwriting recognition (Graves and Schmidhuber 2009; Graves et al. 2009). Unlike other implementations, the CNN did not require any alphabet-specific preprocessing, making it completely general for any language. In the international Arabic recognition competition an accuracy of 91.4% was achieved; the next best implementation was an Arabic-specific rule-based implementation which achieved an accuracy of 87.2%.

Complex tasks involving voluminous data, complex optimisation and function approximation have also benefitted from NN implementations. Medical research has seen great progress in the use of NNs to classify and diagnose complex biological phenomena. Bottaci et al. (1997) used five-year follow-up data from 334 cancer patients to train and validate six NNs designed for the prediction of death within specific periods. The performance of the NNs was measured against that of medical practitioners from various institutions. All six NNs were able to achieve an overall accuracy of greater than 80% for the prediction of death for individual patients.

The recent success of ML has encouraged metal forming researchers to adopt and apply some of the techniques to tackle current challenges. Springback is a common problem in metal sheet forming processes and compromises the accuracy of finished parts. Springback occurs due to the natural elastic recovery of metal after the workpiece is released (Garcia-Romeu, Ciurana, and Ferrer 2009), and has not been reliably captured by Finite Element models (El-Salhi et al. 2012). In a number of studies (El-Salhi et al. 2012; Khan et al. 2015; Kazan, Fırat, and Tiryaki 2009), various geometry representation methods and ML classifiers such as Support Vector Machines (SVMs) and NNs were used to predict springback. When compared with empirical data, the ML classifiers outperformed the FEM models and simple generic classifiers. These springback prediction experiments used small data sets (¡10) and simple geometries as they were comparing their classifier results with manufactured parts.

In manufacturing, however, most of the recent work focuses on quantitative problems such as the prediction of springback mentioned above and cost estimation (Verlinden et al. 2008). Existing work on classifying part geometries tends to be rule-based (Gupta and Gurumoorthy 2013; Kumar et al. 2017) and hence tedious and sensitive to past errors. Additional examples of ML in sheet metal forming applications are detailed in books by Kumar Nee (2017) and Dixit and Dixit (2008). The complex tasks that NNs can tackle make them an appealing choice for the purposes of classifying the geometries produced by metal forming. Similarly, NNs show promise in addressing the challenge of toolpath design in forming processes. Liu et al. (2015) evaluate a variety of NN architectures to optimise parameter control in the Incremental In-Plane Bending process. Opritescu and Volk (2015) train a NN to control a power-hammer device on the basis of desired output shape. In a follow-up paper (Hartmann, Opritescu, and Volk 2016), a more complete framework for such automation is provided.

Geometry description is a fundamental step in many of the papers named above and in any automated version of metal sheet design and manufacturing. There are several methods of converting a 3D geometry into a format suitable for use with a classifier. Many of the geometry representation methods summarise the geometry information as a collection of the main geometric properties as opposed to plain point clouds. In work by El-Salhi et al. (2012), two different geometry representation methods were implemented and compared. The Local Geometry Matrix (LGM) is a global matrix made up of $3 \times 3$ local matrices with the point of interest at the center of the matrix. The values are then converted into qualitative labels to describe the positional information of each point. The LGM method was compared with another method called the Local Distance Mechanism (LDM). This method is based on the observation that the springback tends to be greater further from edges. The LDM describes each grid square center in terms of its distance from its nearest edge. While the author reported high classification accuracy using these geometry description techniques, they are tailored to springback prediction.

Automated feature classification for CAD models is an ongoing area of research and there have been several implementations designed to capture accurately the features of complex 3D geometries. Hegde and Zadeh (2016) designed the FusionNet classifier which used voxel and pixel representations for training relatively weak classifiers; this method was then found to be inferior to using two independent networks for voxel and pixel representations and combining their outputs. Ip (2005) used feature extraction and a location-based descriptor to convert CAD models to numerical arrays. An SVM classifier was then used to compare the similarity and store them in a CAD library. The metal forming processes chosen in the project did not involve any subtractive processes and the generated samples did not contain any distinct and strong features. This meant that feature extraction and classification would not have been directly applicable to some of the geometries.

## 2. Methodology

Data relating to process selection exists in large volumes in the industry, but it is usually proprietary and not readily available. In addition, even when a process is commonly used to produce a specific geometry, that does not ensure it is the optimal choice. Hence, for the purposes of this study, data was generated algorithmically based on textbook and practical knowledge of geometries resulting from specific processes. In the first instance, those geometries were produced as point clouds, which were then described by a variety of metrics and were used as input for training and evaluating the classifier. The various implementations of the classifier tested here consist of NNs. Both the geometry metrics and NN configurations were tested and compared. In this section, the data generation, the geometry metrics and the NN architectures are all presented in detail.

### 2.1. *Data Generation*

The problem of forming a 3D geometry out of a flat sheet of metal is not trivial. Material and geometrical non-linearities make analytical modelling of the deformation intractable and, despite recent improvements in computational speed, numerical simulations remain prohibitively slow for large scale parametric studies. However, in this context, the final geometry of parts can be matched with the corresponding process quite easily once some empirical rules are provided. Five processes with widespread use, distinct deformation characteristics and a large range of product geometries were chosen for this study: Spinning; Deep drawing; Stretch forming; Air bending; Roll bending.

NNs typically require hundreds or thousands of data points to be trained effectively. Hence, the data generation method was designed to produce stochastically representative geometries so that the volume of data points does not require additional rules or labour. Here, the rules corresponding to each process are presented, and typical geometries are shown. Part geometries are generated as point clouds representing surfaces, with no consideration for variation in thickness or material removal. For each of the processes described below, 300 sample geometries were produced, each described by 10,000 points.

#### *2.1.1. Spinning*

Metal spinning is a sheet forming process in which sheet metal is rotated at high speed and formed by imparting sufficient local pressure on the workpiece using a roller or other similar tool. The finished part is axially symmetric with a reduced final diameter relative to the starting diameter (Lange 1985).

Four-point Bézier curves were used to generate the 2D profile of the shape then the profile was revolved around a central axis to generate a 3D surface. The co-ordinates of the Bézier curve were generated stochastically for each sample but were constrained to a specified range to ensure that the geometries had acceptable curvatures. Specifically, the profile of the axisymmetric shape was described by:

$$p(t) = (1-t)^3 \otimes pt_1 + 3(1-t)^2 \otimes pt_2 + 3(1-t)t^2 \otimes pt_3 + t^3 \otimes pt_4 \tag{1}$$

where $\otimes$ is the Kronecker product and the four points defining the curve are:

$$pt_1 = \begin{bmatrix} 0 \\ r_1 \end{bmatrix}, \quad pt_2 = \begin{bmatrix} r_2 \\ r_3 \end{bmatrix}, \quad pt_3 = \begin{bmatrix} r_4 \\ r_3 \end{bmatrix}, \quad pt_4 = \begin{bmatrix} 10 \\ r_5 \end{bmatrix}.$$

The random integers $r_i, i = 1, \ldots, 5$ are in the ranges $(5, 10)$, $(3, 12)$, $(1, 10)$, $(0, 12)$ and $(3, 10)$ respectively, which were chosen to produce realistic geometries. Some typical shapes are shown in Figure 1.

#### *2.1.2. Deep Drawing*

Deep drawing is one of the most widely used sheet forming processes, especially in the automotive industry. In deep drawing, a flat sheet metal blank is formed into a hollow body open on one side by radially drawing the sheet into a forming die by the mechanical action of a punch (Swift and Booker 2003).

Deep drawing is characterised by the variety and complexity of shapes it can generate, including square planforms and complex ridges. It can also generate cylindrical parts similar to metal spinning by using a hollow cylindrical die and punch. To capture these distinct features two separate functions were created for each geometric profile. A function was written to generate 2D circles and extrude them, thus producing cylindrical shapes; the height of the final cylinder and its diameter were generated stochastically. Complex asymmetric shapes were generated starting with two surfaces: one flat and one containing peaks and troughs produced stochastically using trigonometric functions. The two surfaces are then subtracted to produce the final shape, with the surface height given by:

$$z = r_6 - \frac{1}{2}\left(r_{7a}\cos(r_{8a}x/10)^{r_{8b}} + r_{7b}\sin(r_{8c}y)^{r_{8d}}\right), \tag{2}$$

where random integers $r_i, i = 6, \ldots, 8$ are in the ranges $(2, 5)$, $(1, 4)$ and $(-3, 3)$ respectively — these

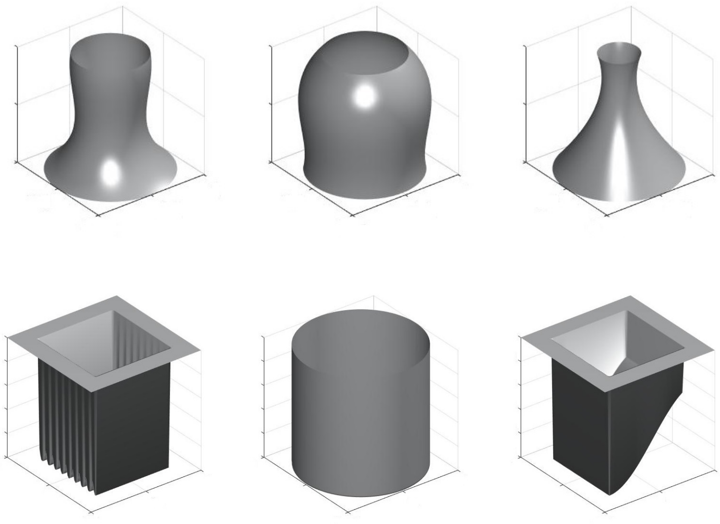

**Figure 1.** *Samples of surfaces generated stochastically to represent the finished geometries of spinning (top) and deep drawing (bottom).*

integers were chosen to produce representative geometries with a rectangular planform. In sum, the surface contained four vertical walls, a flange and random corrugations, or uni-directional curvature. This is not a comprehensive catalogue of deep drawn shapes, but it contains characteristics not seen in shapes produced by other methods. Representative shapes are shown in Figure 1.

### *2.1.3. Stretch Forming*

Stretch forming is a sheet metal forming process in which a sheet of metal is stretched and bent over a die. The sheet is gripped along its edges by gripping jaws which are used to apply tension on the sides to create various geometries (Lange 1985). The curvature and profile of the final parts are changed by using different types of die profiles and varying the amount of tension applied to the work piece from its sides (Swift and Booker 2003).

In order to produce some common geometries, four different cases are noted and reproduced stochastically. All cases have a diminishing curvature towards the boundaries where the sheet is typically clamped. In the first case, a uni-directional parabolic shape is assumed. In the second and third cases parabolic curvature is also assumed in the other direction; positive in the former and negative in the latter (i.e. a saddle shape). For the fourth case, a more general surface is constructed through the use of the exponential function. In analytical form the four cases are:

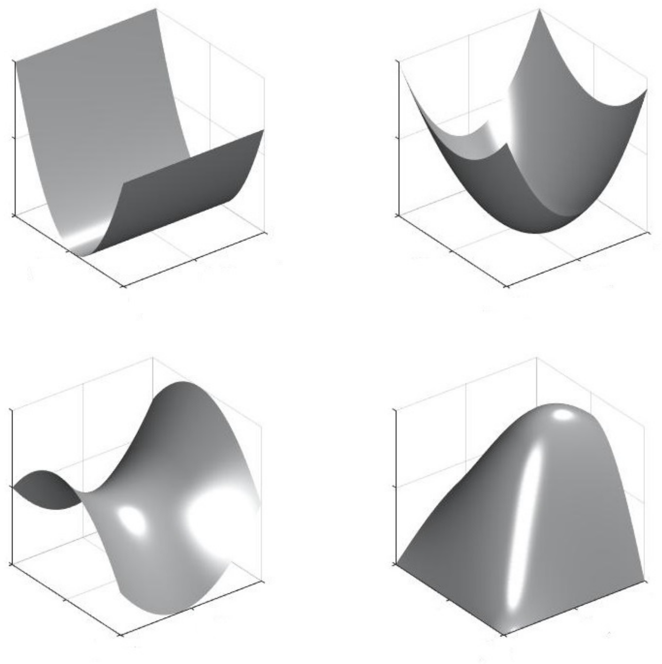

**Figure 2.** *Samples of surfaces generated stochastically to represent the finished geometries of stretch forming.*

$$z = r_9(y - r_{10})^2 \tag{3}$$
$$z = r_9(x - r_{10a}/2)^2 + r_{11a}(y - r_{10b}/2)^2 \tag{4}$$
$$z = r_9(x - r_{10a}/2)^2 - r_{11a}(y - r_{10b}/2)^2 \tag{5}$$
$$z = r_{11a}xy \exp(-r_{11b}x^2 - r_{11c}y^2) \tag{6}$$

where random integers $r_i, i = 9, \ldots, 11$ are in the ranges $(1, 4)$, $(-3, 3)$ and $(1, 5)$ respectively. Samples from all four cases are shown in Figure 2.

### *2.1.4. Air bending*

Sheet metal bending is a widely used and is a vital manufacturing process in industry (Hosford and Caddell 2011). Sheet metal bending causes plastic deformation in one direction to change the geometry of the workpiece (Lange 1985). Various setups using a punch die type setup and fixtures can be utilised to produce different bending radii and control the amount of springback (Swift and Booker 2003).

Four-point Bézier curves were used to generate this category of geometries. The 2D profile of the shape was initial defined and then reproduced along the third dimension to create a 3D surface. The key features here were the localised nature of the deformation, and the presence of resulting curvature only in one direction. With reference to Equation 1, the following points were used to define the profile shape:

$$pt_1 = \begin{bmatrix} 0 \\ 10 \end{bmatrix}, \quad pt_2 = \begin{bmatrix} 6 - r_{12} \\ r_{13} \end{bmatrix}, \quad pt_3 = \begin{bmatrix} r_{12} \\ r_{13} \end{bmatrix}, \quad pt_4 = \begin{bmatrix} 6 \\ 10 \end{bmatrix}.$$

where random integers $r_i, i = 12, 13$ are in the ranges $(0, 3)$ and $(0, 5)$ respectively.

#### *2.1.5. Roll Bending*

Roll bending provides a simple technique that is useful for bending sheets continuously along their length (Lange 1985). Roll bending involves the use of three rolls to feed and bend the sheet to desired curvatures. Different profiles can be obtained by controlling the distance and relative angle of the rolls. Actuated rolls can be used to control the curvature along the length of the sheet (Taylan 2012).

To generate similar geometries four-point Bézier curves were used to generate the 2D profile of the shape and then extruded to produce a 3D surface. The extent of extrusion and points of the defining Bézier curve are generated stochastically within given constraints.

$$pt_1 = \begin{bmatrix} 0 \\ 10 \end{bmatrix}, \quad pt_2 = \begin{bmatrix} 10 - r_{14a} \\ r_{14b} \end{bmatrix}, \quad pt_3 = \begin{bmatrix} r_{14a} \\ r_{14b} \end{bmatrix}, \quad pt_4 = \begin{bmatrix} r_{15} \\ r_{14c} \end{bmatrix}.$$

where random integers $r_i, i = 14, 15$ are in the ranges $(0, 10)$ and $(15, 25)$ respectively.

#### *Testing set*

The data generation for testing sets used the same functions as above but with variation on the constrain parameters to ensure "unknown" geometries were included. The random testing set generation used the same definitions as for the training set, but was made distinct by using the MATLAB `rng()` function.

The classification accuracy of each network was calculated as the percentage of correct classifications in the total number of samples in the testing dataset. Another important measure was the average margin of confidence of the classifications. To identify the limitations of each geometry representation, the misclassifications were counted for each manufacturing process. The training and testing of different NNs is detailed in Section 3.

### 2.2. *Geometry Representation*

To input each geometry into the classifier, a geometry representation method is required to extract its unique features and present them as a matrix of numerical values. Manufacturing processes produce finished parts with distinctive curvature properties as described in the previous section. Geometry representation methods which describe the curvature of the surface at each point can capture small differences between the different finished parts of various processes. Four different geometry representation methods which aim to fully capture the characteristics of the surface were tested and their effect on the performance of the classifier was compared.

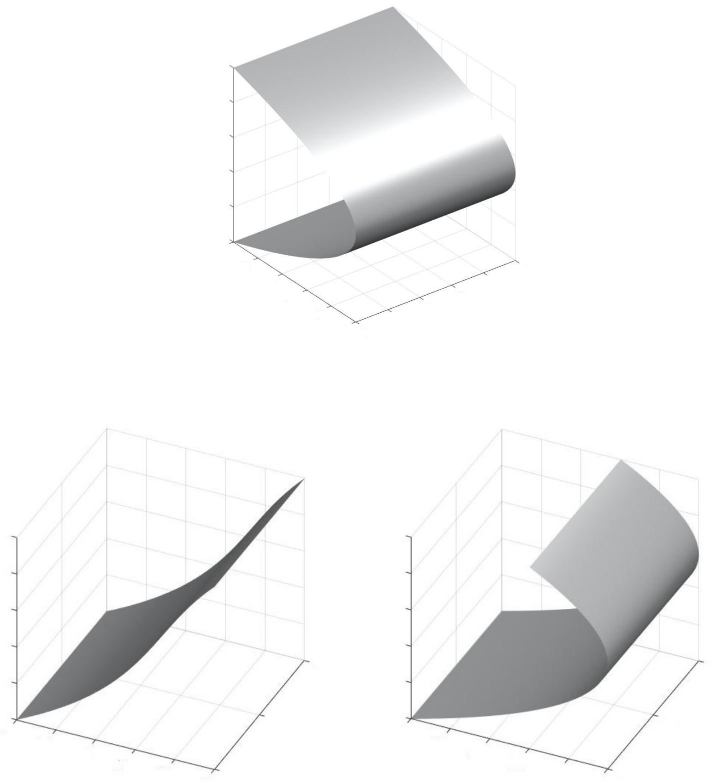

**Figure 3.** *Samples of surfaces generated stochastically to represent the finished geometries of air bending (top) and roll bending (bottom).*

### 2.2.1. *Principal, Mean, and Gaussian Curvature*

The principal curvatures for a given surface describe how the surface bends by different amounts in different directions at each point (Guggenheimer 1977). By calculating the principal curvatures at every point of a 3D surface, a 2D representation of the geometry can be generated in order to describe fully the curvature to the classifier. For a given 3D differentiable surface in Euclidean space, a unit normal vector can be drawn at each point on the surface. The normal plane will contain a normal vector and a tangent vector to the surface which cuts the surface in a plane curve. This curve will have different curvatures for different normal planes at each point. The principal curvatures for each point are denoted as $\kappa_1$ and $\kappa_2$ and refer to the maximum and minimum values of this curvature.

The Mean and Gaussian curvatures are quantities derived from the principal curvatures $\kappa_1$ and $\kappa_2$ and can be calculated for each point on the surface (Pressley 2010). The Mean curvature, $H$, is the arithmetic mean of the principal curvatures:

$$H = \frac{\kappa_1 + \kappa_2}{2}$$

while the Gaussian curvature, $K$, is the square of their geometric mean:

$$K = \kappa_1 \kappa_2$$

The three quantities defined here — the principal curvatures; the Mean curvature; the Gaussian curvature — were used as inputs to the classifier in this study. A fourth representation was created by combining the Mean and Gaussian curvatures. Figure 4 gives a visual representation of how 3D surfaces are processed using the geometry representation methods for use with the classifier. Starting from the point cloud representation, the quantities above were calculated for points along a 3D grid and an appropriate 2D representation of the geometry was created. This served as an input to the NN. Multiple databases for each method were created to compare their efficacy. The method of representation resulting in the highest accuracy was used in the final implementation.

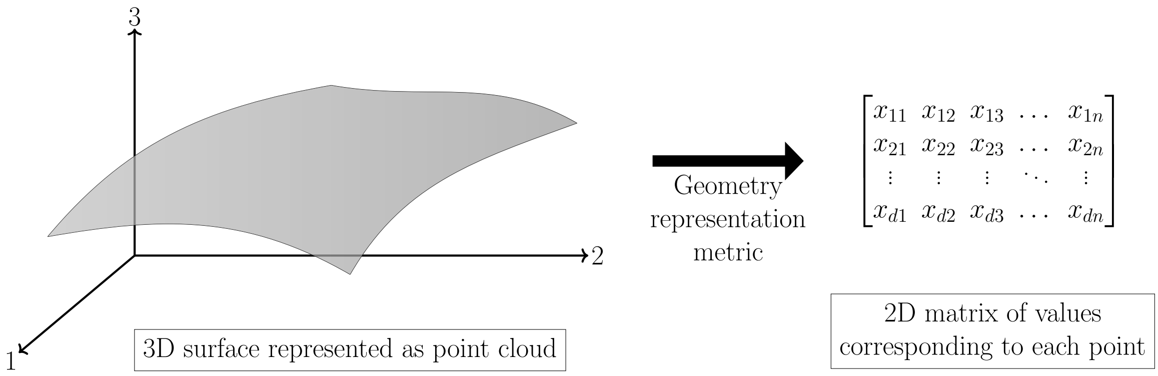


**Figure 4.** Flowchart detailing how the 3D surfaces are converted into a usable input for the classifier using the geometry representation methods.

### 2.3. *Neural Network Architecture*

Two different NN architectures were used: a shallow single layer NN and a multi-layer NN with several specialised layers. A simple NN architecture is shown in Figure 5, with a small number of nodes shown for clarity. All NNs were constructed and tested within MATLAB R2017a using the Neural Network Toolbox.

The single layer NN is computationally efficient and is relatively simple to implement. However, it offers very little flexibility in terms of the input. A single layer classifies the inputs based on their absolute magnitude, relative magnitude and position within the input space. The input format is simply a 1D array corresponding to the extracted geometric features of each point on the surface.

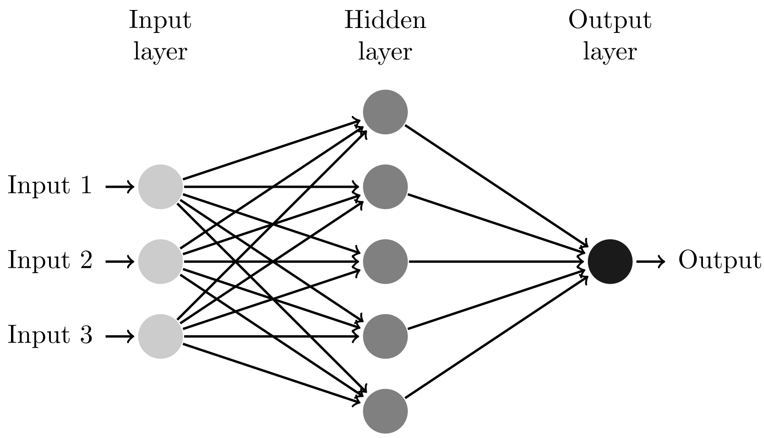


**Figure 5.** *A diagram of a simple NN architecture with three input nodes one hidden layer with five nodes.*

On the other hand, a deep CNN implementation uses several specialised layers which are stacked to transform and manipulate the input to extract features. Each layer has a specialised function and DNNs use multiple layers to increase the variety of features found. These can enable scale and orientation independent detection of features when convolutional layers are added. Each layer represents a function that transforms an input and passes it onto the next layer. The five layers used here are listed with a brief explanation for each one:

- Convolution layer: This layer applies convolutional filters to the input. By splitting the input into subregions, mathematical operations can be performed on local regions of the input to produce a single value corresponding to a specific feature at a specific location in the input.
- ReLU layer: The ReLU layer applies an activation function to produce non-linearities in the model.
- Max pooling layer: The max pooling layer downsamples the input to reduce dimensionality by keeping the maximum value of different subregions and discarding all other values. Max pooling is critical for DNNs as it decreases processing time.
- Fully connected layer: Dense (fully connected) layers perform classification on the features extracted and down-sampled by the convolutional and pooling layers.
- Softmax layer: The softmax function is usually the last layer of a NN-based classifier and is used to provide a probability for each possible label in the output layer.

More details on the purpose and function of each layer can be found in Beale, Hagan, and Demuth (2017). Three configurations were trained and tested for CNNs. They feature variations in number and placement of layers and convolution layer size. The three different structures are detailed in corresponding diagrams in Figure 6.

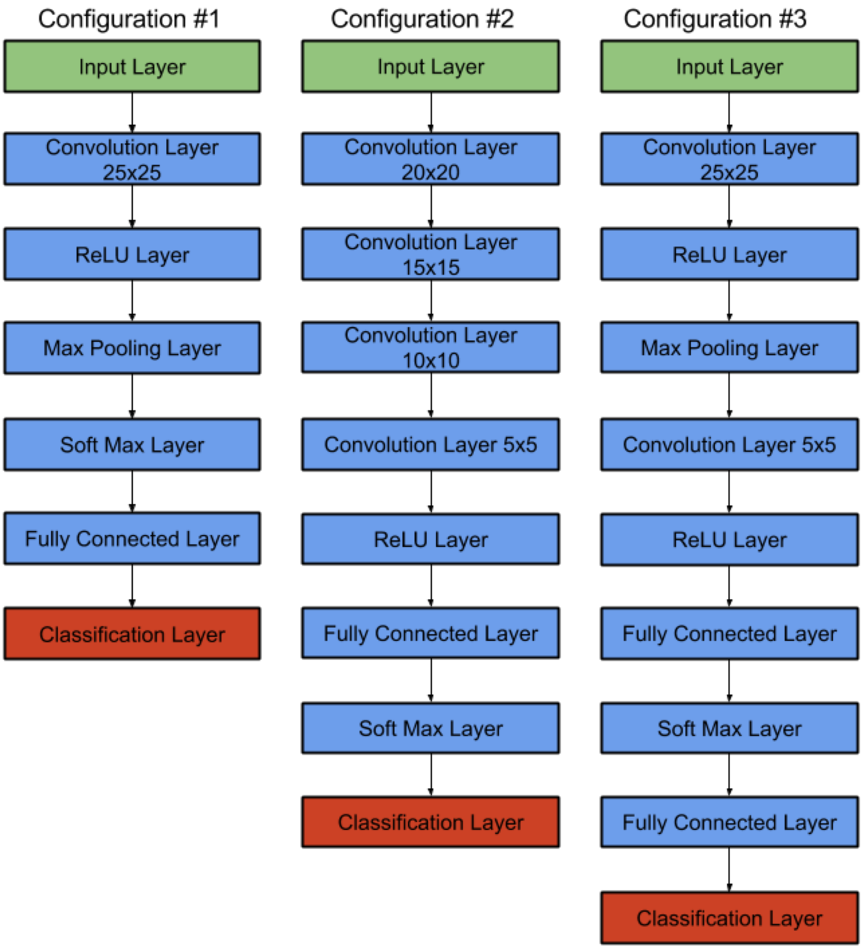


**Figure 6.** *Image detailing the configuration of the three deep CNNs used for classification.*

## 3. Results

Both the NN architectures and the geometry representations were tested using the testing set data from Section 2.

### 3.1. *Single Layer Neural Networks*

The two main parameters controlling the configuration of the single layer NN are the hidden layer size and the number of samples per process constituting the training data set. These parameters were varied independently to gain an understanding of their effect on NN classification performance. The size of the hidden layer was varied between 10 and 100 nodes and the classification accuracy of the network was tested on a fixed testing sample set. Training of the single layer NNs took 1-10 minutes on a quad core 2.7GHz machine with 16GB of RAM, depending on the variation in input parameters. It can be seen in Figure 7 that the best performance is achieved at a hidden layer size of 50 for the principal and Gaussian curvature implementations. As the hidden layer size is increased the performance decreases gradually or plateaus. The other two representation methods have a statistically insignificant classification accuracy at all hidden layer sizes.

The size of the data set might also affect classification accuracy, hence the number of samples per process used for training was varied and the classification accuracy was recorded for each database size and is presented in Figure 8. Results for training set sizes between 10 and 200 geometries per process are included. The principal curvature method considerably outperforms the other geometry representation methods and performs much better when trained on the same data set. The Gaussian and Mean curvatures combination method fails to achieve statistically significant results.

Increasing the size of the training data set positively affects the classification performance. Furthermore, there appears to be a critical dataset size at which the accuracy of the classifier is greatly improved; this is likely to be the point at which there is enough variation between samples for each process to allow for distinct classification. Figure 8, which gives the results for the principal curvature line, evidences this. Here, large increases in accuracy are seen when the number of training samples exceeds 100 per process. One possible explanation for the poor performance of the combined Gaussian and Mean curvature method might be that its critical dataset size is greater than 200 samples per process.

While this type of sensitivity analysis helps with understanding the relationship between the inputs and outputs of the NN, as the hidden layer size and volume of training data is increased, further challenges arise due to computational and time constraints. To assess the performance of the simple NN for each geometry representation method, the hidden layer size was fixed at a size of 50 and a training dataset of 5000 samples was used for each geometry representation method resulting in 4 different classifiers. The classification performance of each of the classifiers is summarised in Table 1. The highest level of accuracy, 79.3%, was achieved using the principal curvature method. The Mean and Gaussian curvatures combination do not produce acceptable classification accuracies.

All the methods have a low confidence margin ($< 30\%$) in comparison to some of the NNs found in the literature such as in Simonyan and Zisserman (2014) and Szegedy et al. (2015), where confidence margins over 60% are achieved. Low confidence margins mean that the classifier fails to capture all the differences between the geometries produced by different methods. The high accuracy combined with low confidence margins implies that the simple NN is adjusting the weights of the hidden layer based on the magnitude of the values detected in the input. Deeper NNs which use more layers, convolution and max pooling will enable the classifier to capture additional geometric features. For example, bending and deep forming can both produce sharp corners. To distinguish between them the classifier must be able to interpret the positional information of the corners as well as detect their presence. Convolution enables this process by using sliding windows which act as local filters to identify features and their locations.

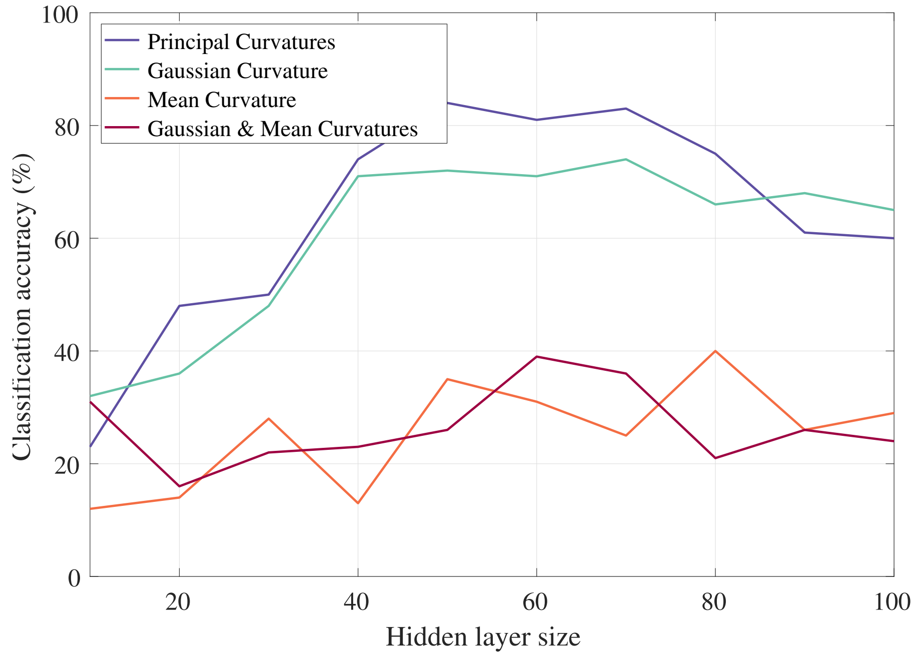


**Figure 7.** *Classification accuracy of the single layer NN versus the hidden layer size with a fixed training data set size of 50 samples per process.*

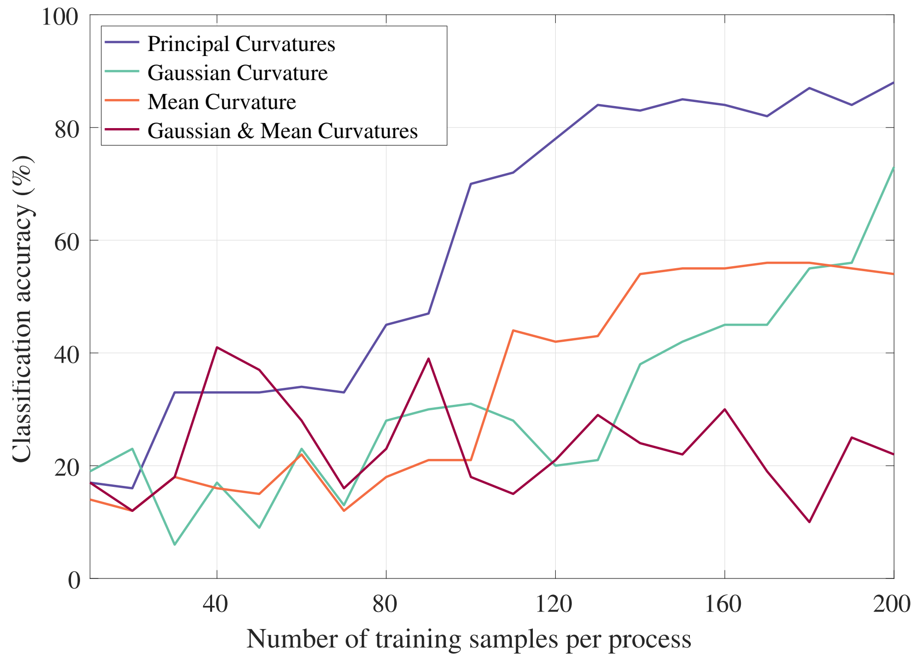


**Figure 8.** *Classification accuracy of the single layer NN versus the number of training samples used per process with a fixed hidden layer size of 50.*

**Table 1.** Summary of the performance of the single layer NN using different geometry representation methods.

| Geometry representation method | Accuracy (%) | Confidence margin (%) |
|---|---|---|
| Principal curvatures | 79.3 | 26.2 |
| Gaussian curvature | 73.4 | 21.3 |
| Mean curvature | 54.2 | 23.7 |
| Gauss. & Mean curvatures | 41.3 | 24.8 |

### 3.2. *Deep Convolutional Neural Networks*

The three different deep CNN configurations were tested and compared using the same data and methods described in Sections 2.1 and 2.2. Each configuration shown in Figure 6 consists of an arrangement of layers which perform a specific function. Section 2.3 contains a brief description of the purpose of each layer.

Typically, an increase in the size of the training set used with CNNs improves the quality of the classifier. A training data set of 10k samples was built consisting of 2k samples per process for each geometry representation method. The final training database consisted of 40k samples. Training each NN configuration took around 4 hours on a quad core 2.7GHz machine with 16GB of RAM. Each CNN configuration for each geometry representation method was tested using the testing framework outlined in Section 2.1.5.

The results obtained from the testing are summarised in Table 2. Using a deeper CNN only improves the accuracy of the classifier by a small margin. However, it was observed that each additional layer increased the training duration by about 10%. The deepest CNN Configuration #3 shown in Figure 6 outperformed the other two configurations for all of the geometry representation methods except the Mean curvature method. The combined Gaussian and Mean curvature method of geometry representation outperforms all others by a significant margin, greater than 10%. With this geometry representation and with Configuration #3 for the classifier, the best classification accuracy of 88.8% was achieved. This was not the case using the simple NN, indicating that the single layer approach fails to capture all of the features present in a combined Gaussian and Mean curvature representation method. Additional layers for the CNNs using principal curvature did not yield any improvement in the accuracy of the classifier.

NNs can be frustrating as their predictions often lack traditional rigour. The weights the NN self-assigns during training aim to approximate a specific function; unfortunately, the link between these weights and the function are not direct, clear or easily presented. “Confusion matrices” are sometimes used to offer some qualitative insight into the NN’s inner workings. These allow a direct visual comparison of performance. Here, such a matrix was produced to further understand the limitations of the geometry representation method. This confusion matrix was produced for each geometry representation method using CNN Configuration #3. These results are presented in Figure 9. The diagonal of each confusion matrix represents the percentage of correct classifications of each process. Other cells represent the percentage that the correct process was misclassified for another process.

Pure bending appears to be the easiest process to classify with 100% classification accuracy for all geometry representation methods. However, most of the misclassifications were labelled as bending, suggesting that the training data set has samples which skew the classifier. Using the Gaussian curvature method alone to classify roll bending leads to a 0% accuracy indicating that the Gaussian curvature method fails to capture the key resultant geometric features of the process when compared to the other geometry representation methods. This was anticipated since for uni-directional curvature, the GC is trivially zero throughout the surface.

The combined Gaussian and Mean curvature method shows synergistic effects and achieves higher accuracies relative to the use of the two metrics separately. Higher accuracies are seen when classifying deep forming, stretch forming and roll bending using the combination method. It appears that combining the Gaussian and Mean curvature methods does not allow them to fully exploit their relative successes for all of the processes. For example, the Gaussian curvature method does well in classifying spinning 80% of the time, whereas the mean curvature method only achieves a correct spinning classification accuracy of 15%. When the metrics are combined, a 66% accuracy rate is seen, moderating the results of individual metrics.

The confidence margins obtained using the deep networks are much higher — as shown in Table 2 — in comparison to the values obtained using the simple NN in Table 1. The confidence margins are at least 30% higher for all geometry representation methods indicating that adding more layers, convolution and reLU aids with capturing additional features. Higher confidence margins mean that the predictions of the classifier are based on clearer, distinct features which have been detected in each sample. The Gaussian and Mean curvature methods used in isolation produce the lowest confidence margins indicating a deficiency capturing all the geometric features required for accurate classification.

**Table 2.** Summary of the performance of each deep NN configuration using different geometry representation methods

| | **Configuration #1** | | **Configuration #2** | | **Configuration #3** | |
|---|---|---|---|---|---|---|
| **Geometry representation method** | **Accuracy (%)** | **Confidence margin (%)** | **Accuracy (%)** | **Confidence margin (%)** | **Accuracy (%)** | **Confidence margin (%)** |
| Principal curvatures | 73.2 | 79.9 | 78.9 | 81.6 | 77.6 | 81.0 |
| Gaussian curvature | 63.3 | 61.3 | 65.6 | 48.1 | 68.0 | 60.3 |
| Mean curvature | 65.7 | 56.3 | 66.6 | 48.3 | 64.2 | 64.6 |
| Gauss. & Mean curvatures | 85.6 | 82.0 | 86.3 | 72.2 | 88.8 | 87.6 |

## 4. Discussion

The best performing geometry representation method was found to be the combined Gaussian and Mean method. There are more geometry representation methods that can be explored to further improve accuracy. One suggestion is to use a combination of the principal curvature, Mean and Gaussian curvatures in one matrix even though the latter two are derived values. Another possible route would be to focus solely on principal curvature and use a more sophisticated network structure to capture the derived quantities without computing them. Methods which can also account for changes in the volume of the final geometry would open up the ability to classify subtractive manufacturing processes if the initial shape is known. The work done by Hegde and Zadeh (2016) suggests that using multiple NN for representing the same sample in different ways could be more effective than using one NN with combined geometry representation methods.

Using a multi-layered CNN greatly improved the accuracies of the more complex geometry representation methods and boosted confidence margins. The best configuration was a deep CNN, using

| **Principal Curvatures** | Spinning | Deep Drawing | Stretch Forming | Air Bending | Roll Bending |
|---|---|---|---|---|---|
| Spinning | 90% | 0 % | 0% | 9% | 1% |
| Deep Drawing | 0% | 52% | 0% | 48% | 0% |
| Stretch Forming | 18% | 0% | 76% | 0% | 6% |
| Air Bending | 0% | 0% | 0% | 100% | 0% |
| Roll Bending | 7% | 0% | 0% | 47% | 46% |

| **Gaussian Curvature** | Spinning | Deep Drawing | Stretch Forming | Air Bending | Roll Bending |
|---|---|---|---|---|---|
| Spinning | 80% | 0 % | 1% | 19% | 0% |
| Deep Drawing | 0% | 92% | 0% | 5% | 3% |
| Stretch Forming | 0% | 2% | 53% | 45% | 0% |
| Air Bending | 0% | 0% | 0% | 100% | 0% |
| Roll Bending | 0% | 0% | 0% | 100% | 0 % |

| **Mean Curvature** | Spinning | Deep Drawing | Stretch Forming | Air Bending | Roll Bending |
|---|---|---|---|---|---|
| Spinning | 15% | 0 % | 0% | 84% | 1% |
| Deep Drawing | 0% | 46% | 0% | 54% | 0% |
| Stretch Forming | 4% | 0% | 87% | 5% | 4% |
| Air Bending | 0% | 0% | 0% | 100% | 0% |
| Roll Bending | 0% | 0% | 0% | 53% | 47% |

| **Gauss./Mean Curvatures** | Spinning | Deep Drawing | Stretch Forming | Air Bending | Roll Bending |
|---|---|---|---|---|---|
| Spinning | 66% | 0 % | 1% | 33% | 0% |
| Deep Drawing | 0% | 100% | 0% | 0% | 0% |
| Stretch Forming | 0% | 0% | 91% | 3% | 6% |
| Air Bending | 0% | 0% | 0% | 100% | 0% |
| Roll Bending | 0 % | 0% | 0% | 49% | 51% |

**Figure 9.** *"Confusion matrices" produced by using CNN Configuration #3. The vertical axis represents the actual process presented to the classifier and the horizontal axis represents the process predicted by the classifier.*

the Gaussian and Mean curvature representation method and achieved an accuracy of 88.8% with an average confidence margin of 87.6%. The convolution filter should also improve predictions when the classifier is faced with geometries which are very different relative to the training samples since it can handle scaling and rotation of features. Tracing the source of the resulting misclassifications is problematic as they can arise from several factors, including incomprehensive or unrepresentative training data, insufficient training of the NN and misrepresentation of the sample due to the chosen geometry representation method. The greatest disadvantage of using NNs is the lack of tractable explanation for the observed results.

Measuring the performance of the classifier is further hindered by the nature of the data used here for training and testing. A better approach would be to use curated data sets from industry. The data format of NNs becomes inflexible post training as the input has to be in the same format of the training data set. This manifests constraints on the size and shape of the data. Such a limitation can be problematic in practice. If the input to the classifier is a CAD file it would require some manipulation such as down-sampling to work within these constraints. However, the methods for such data transformations are available.

The computation time for training was typically more than 20 times longer for the CNNs than for the simple NNs, while in absolute terms the training duration ranged from minutes to hours. However, training is a sporadic activity that does not happen online. Hence this requirement should not be a deterrent for practical applications. In addition, the tests were performed on a standard PC and not on dedicated hardware, which could improve performance. Once the NNs were trained, the computation time to produce a classification in response to a geometry is orders of magnitude smaller.

A serious limitation of the current implementation is the inability to deal with geometries that resulted from multiple processes. It is typical in industry that finished parts have gone through more than one forming process. Additional work would be required to identify the multiple manufacturing processes and the order of their application to produce a finished part. This adds a significant amount of complexity to the classification process and would require a more comprehensive training data set.

Choosing a suboptimal process can incur considerable costs in the long run, especially with mass production applications. A high degree of trust in the classifier is required in order to base manufacturing decisions on its output. This trust is unlikely without empirical proof and real use cases demonstrating its sound ability.

NN-based classifiers have seen great success when applied to complex and voluminous applications such as image recognition and financial trading. In such applications, the volume of work makes the tasks impossible for a human to achieve. In the case of classifying manufacturing processes, the task is fairly complex. However, it might not be sufficiently voluminous to justify the cost of developing such a tool. One exciting solution would be to encourage separate manufacturers to build a collective labelled training database of their manufacturing processes. This would quickly produce a large training data set, and a more capable classifier. One way around the data security and integrity of this feature is to use CryptoNets, a modified version of NNs which can be applied to encrypted data Gilad-Bachrach et al. (2016). This approach allows users to provide training data in an encrypted form ensuring that it remains confidential.

### 4.1. *Classification and Selection*

The ML classifiers detailed above return the likelihood that an input geometry was produced by one of the five processes. In practice, the process with the highest likelihood could be selected as the most suitable method for forming the particular part. Additional parameters beyond the geometry could be included in the training data so as to improve the accuracy of the algorithm, while additional processes can also be represented to make the selection more comprehensive. The output data are of use beyond selection since they reveal which processes are related. One scenario of this is when two processes are equally likely to produce a certain geometry. Similarly, when all known methods are equally likely to have produced a given geometry, a gap in the manufacturing range is identified and, in turn, a new process design becomes necessary.

## 5. Conclusions

In this study, NN classifiers were constructed for the first time to demonstrate its usefulness in automating and enhancing the sheet forming selection process. Automation of this aspect of design and manufacturing has only been considered with rule-based programs. The high accuracy rates achieved with a deep CNN suggest that rule-based implementations can be replaced with ML methods. The results show how the ML methods which have been successful in other applications such as facial recognition can be used to classify manufacturing processes based on only the final geometry as an input. Four different geometry representation approaches were compared, with a combination of Mean and Gaussian Curvatures producing the best results. This work represents a viable new application for Machine Learning within Manufacturing.

Such a classifier could be packaged into a software tool to aid design and manufacturing engineers. Designers can use the tool to predict the manufacturing process needed to realise their CAD models, especially when human expertise is not available. In turn, early cost estimates can be produced. Manufacturing engineers can use the tool as an aid to choose suitable processes and explore options which are not immediately obvious to them. The same approach can assist researchers in the automatic design of new forming processes (Loukaides and Allwood 2016; Music and Allwood 2012) by showing the statistical distance between existing methods. Shapes that do not clearly correspond to one forming process either indicate capabilities overlap or that there is a gap. This project was limited to five sheet forming processes, but the same approach can be used to generate training data sets which cover more processes and expand the functionality of the classifier.

## References


Beale, Mark Hudson, Martin T. Hagan, and Howard B. Demuth. 2017. *Neural Network Toolbox$^{TM}$ User's Guide, R2017a.* The MathWorks, Inc.

Bottaci, Leonardo, Philip J Drew, John E Hartley, Matthew B Hadfield, Ridzuan Farouk, Peter WR Lee, Iain MC Macintyre, Graeme S Duthie, and John RT Monson. 1997. "Artificial neural networks applied to outcome prediction for colorectal cancer patients in separate institutions." *The Lancet* 350 (9076): 469–472.

Dempster, Michael A. H., and Vasco Leemans. 2006. "An automated FX trading system using adaptive reinforcement learning." *Expert Systems with Applications* 30 (3): 543–552.

Dixit, Prakash Mahadeo, and Uday Shanker Dixit. 2008. *Modeling of metal forming and machining processes: by finite element and soft computing methods.* Springer Science & Business Media.

El-Salhi, Subhieh, Frans Coenen, Clare Dixon, and M Sulaiman Khan. 2012. "Identification of Correlations Between 3D Surfaces Using Data Mining Techniques: Predicting Springback in Sheet Metal Forming." In *SGAI Conf.*, 391–404. Springer.

Garcia-Romeu, M. L., J. Ciurana, and I. Ferrer. 2009. "Springback determination of sheet metals in an air bending process based on an experimental work." .

Gilad-Bachrach, Ran, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. 2016. "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy." In *International Conference on Machine Learning*, 201–210.

Graves, Alex, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. 2009. "A novel connectionist system for unconstrained handwriting recognition." *IEEE transactions on pattern analysis and machine intelligence* 31 (5): 855–868.

Graves, Alex, and Jürgen Schmidhuber. 2009. "Offline handwriting recognition with multidimensional recurrent neural networks." In *Advances in neural information processing systems*, 545–552.

Guggenheimer, H. W. 1977. *Differential geometry.* McGraw-Hill, New York.

Gupta, Ravi Kumar, and Balan Gurumoorthy. 2013. "Classification, representation, and automatic extraction of deformation features in sheet metal parts." *Computer-Aided Design* 45 (11): 1469–1484.

Hartmann, Christoph, Daniel Opritescu, and Wolfram Volk. 2016. "An artificial neural network approach for tool path generation in incremental sheet metal free-forming." *Journal of Intelligent Manufacturing* 1–14.

Hegde, Vishakh, and Reza Zadeh. 2016. "FusionNet: 3D object classification using multiple data representations." *arXiv preprint arXiv:1607.05695* .

Hosford, William F, and Robert M Caddell. 2011. *Metal forming: mechanics and metallurgy.* Cambridge University Press.

Ip, Cheuk Yiu. 2005. "Automatic classification of CAD models." Master's thesis, Drexel University.

Kazan, Recep, Mehmet Fırat, and Aysun Egrisogut Tiryaki. 2009. "Prediction of springback in wipe-bending process of sheet metal using neural network." *Materials & design* 30 (2): 418–423.

Khan, Muhamad S, Frans Coenen, Clare Dixon, Subhieh El-Salhi, Mariluz Penalva, and Asun Rivero. 2015. "An intelligent process model: predicting springback in single point incremental forming." *The International Journal of Advanced Manufacturing Technology* 76 (9-12): 2071–2082.

Kumar, Navdeep, and Pankaj Garg. 2011. "Recognition of distorted CAD objects using neural networks." *International Journal of Computer Applications* 14 (8): 18–22.

Kumar, Shailendra, Rajender Singh, Deepak Panghal, Sachin Salunkhe, and Hussein MA Hussein. 2017. "Feature Extraction and Manufacturability Assessment of Sheet Metal Parts." In *AI Applications in Sheet Metal Forming*, 41–66. Springer.

Lange, Kurt. 1985. *Handbook of metal forming.* McGraw-Hill.

Li, Stan Z., and Juwei Lu. 1999. "Face recognition using the nearest feature line method." *IEEE transactions on neural networks* 10 (2): 439–443.

Liu, Nan, He Yang, Heng Li, Siliang Yan, Honglie Zhang, and Wenting Tang. 2015. "BP artificial neural network modeling for accurate radius prediction and application in incremental in-plane bending." *International Journal of Advanced Manufacturing Technology* 80.

Loukaides, EG, and JM Allwood. 2016. "Automatic design of sheet metal forming processes by un-forming."

*International Journal of Mechanical Sciences* 113: 61–70.

Music, Omer, and Julian M Allwood. 2012. “The use of spatial impulse responses to characterise flexible forming processes with mobile tools.” *Journal of Materials Processing Technology* 212 (5): 1139–1156.

Nee, A.Y.C. 2017. “An Overview of Applications of Artificial Intelligence (AI) in Sheet Metal Work.” In *AI Applications in Sheet Metal Forming*, 1–13. Springer.

Opritescu, Daniel, and Wolfram Volk. 2015. “Automated driving for individualized sheet metal part productionA neural network approach.” *Robotics and Computer-Integrated Manufacturing* 35: 144–150.

Pressley, Andrew N. 2010. *Elementary differential geometry*. Springer Science & Business Media.

Simonyan, Karen, and Andrew Zisserman. 2014. “Very deep convolutional networks for large-scale image recognition.” *arXiv preprint arXiv:1409.1556* .

Swift, K. G., and J. D. Booker. 2003. *Process selection: from design to manufacture*. Butterworth-Heinemann.

Swift, K. G., and J. D. Booker. 2013. *Manufacturing process selection handbook*. Butterworth-Heinemann.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. “Going deeper with convolutions.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

Taylan, Altan. 2012. *Sheet Metal Forming Fundamentals*. Materials Park.

Verlinden, Bart, JR Duflou, Philippe Collin, and Dirk Cattrysse. 2008. “Cost estimation for sheet metal parts using multiple regression and artificial neural networks: A case study.” *International Journal of Production Economics* 111 (2): 484–492.

Widodo, Achmad, and Bo-Suk Yang. 2007. “Support vector machine in machine condition monitoring and fault diagnosis.” *Mechanical systems and signal processing* 21 (6): 2560–2574.