




An Approach for Integrating Analytical and Experiential Knowledge for Structural Design

Satchit Ramnath¹ , Konrad Witek², Jami J. Shah³, Duane Detwiler⁴

¹Clemson University, sramnat@clemson.edu

²The Ohio State University, witek.9@osu.edu

³The Ohio State University, shah.493@osu.edu

⁴Honda Research Institute, USA, ddetwiler@honda-ri.com

Corresponding author: Satchit Ramnath, sramnat@clemson.edu

Abstract. The development of Topology Optimization methods has enabled engineers to generate shapes that satisfy given functional and structural constraints. These methods are derived from mechanics principles embedded in FEA. They progressively remove underutilized material and keep material that represents load paths. However, they produce organic shapes that may be non-manufacturable and are not easily converted to parametric CAD models for size optimization. Consequently, many designers start with proven past designs that may already be in the form of parametric CAD and modify them to new specifications. The first approach is purely analytical, while the second uses experiential knowledge. This paper presents a methodology to combine the advantages of both, as follows. Two large data sets are artificially curated, one generated from existing designs (parametric CAD) and the second from "fuzzification" of those designs to resemble features in topology-optimized meshes. An algorithm is trained to match feature patterns of a given TopOpt mesh to members of the fuzzy data set. Now, assume that the designer starts by performing topology optimization of a new component of the same type. The trained algorithm is used to find the closest matches of that design in the fuzzy set, which relates to the seed design in the parametric data set. Finally, the feature pattern found is conformal mapped to the geometry of the new design. While this methodology is primarily demonstrated using automotive hood frames, it can be generalized to any sheet metal component. The last section discusses the generalizability of this approach.

Keywords: topology optimization, structural design, applied machine learning, data-driven design, conformal mapping

DOI: <https://doi.org/10.14733/cadaps.2025.42-67>

1 INTRODUCTION

In the design of mechanical components and support structures, one must perform stress and failure analysis. Function constrains some shape and size parameters; the rest are determined by minimizing volume/mass while keeping the structure within safe strength limits. The domain of this study is automotive upper-body design. Designing an automotive structure is a multilayered complex problem involving consideration of operational and crash safety, cost, regulations, weight, noise and manufacturability. Upper body frames are essentially welded sheet metal assemblies (Fig. 1). Components are commonly stamped from 1.2 mm DP steels. The load-bearing structure is severely constrained between the inner and outer styling surfaces.

The component chosen for this study is the structural design of the support frame for hoods. Hood skin surfaces vary in width, length, aspect ratio, normal depth and curvatures; outer edges have various profiles, and some have waterfall sides. Figure 2a shows various rib and pocket feature patterns. Frame characteristics are dictated by styling; the hood designer needs to come up with the rib/pocket pattern for the hood frame and secondary features, such as hinges, locks, etc. (Fig. 2b) The purpose of ribs is to add bending stiffness, while pockets are for reducing weight. These designs have evolved by trial and error over many generations of vehicles rather than by pure analytical methods. Of course, any such design needs to go through stress and failure analyses and testing.



Figure 1: Upper body structure

When a new vehicle model is developed, the hood styling surfaces may change in size, aspect ratio and curvatures. The common tendency then is to adapt the previous feature pattern to the new skin. Past designs encode valuable experiential knowledge from analysis, testing and consumer field use data. Although time efficient and safe, this approach may be missing opportunities for producing new shapes or sizes. Additionally, feature patterns not explored before for a given vehicle platform may lead to meeting objectives better. The advent of Topology Optimization (TopOpt or TO) has made it possible to produce new shapes from first principles to satisfy given boundary conditions, mass minimization objectives and constraints. However, these organic shapes seem difficult to manufacture at first sight by mainstream mass production processes. These observations lead to the following question: What if a very large data set of hood frames existed, containing a

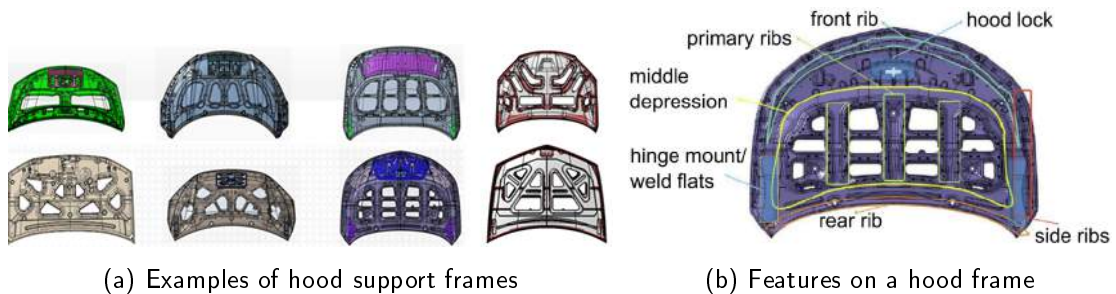


Figure 2: Hood support frames

large variety of feature patterns, shapes and sizes; could the irregular organic designs produced by TopOpt be used to find a close match in the hood frame database? This is the central question that is explored in this paper.

2 TECHNOLOGY REVIEW

Three areas relevant to this work will be reviewed here: **1)** manufacturing constraints embedded in extant topology optimization software, **2)** large geometric data sets for ML, and **3)** machine learning algorithms. The methods proposed in this paper are novel and literature on a combination of methods is very limited. This section discusses literature relevant to the tools and algorithms used to develop and test the proposed methodology.

2.1 Topology Optimization and Manufacturing Constraints

To extend the manufacturability of TopOpt designs beyond additive manufacturing, state-of-the-art software has introduced so-called manufacturing constraints. These include min/max member size, symmetry about given planes, uniform cross-section in a particular direction to make a part amenable to extrusion and tapering in some direction apparently to allow removing from dies and molds [30]. These are rather simplistic attempts at manufacturability. For example, many casting processes may involve multiple-piece dies/molds, cores, ejector pins, etc., not just unidirectional ejection. Also, TopOpt today cannot produce hollow sections, such as those formed by joining two stamped hat sections at flanges, like that in A and B pillars. Another issue is producing pockets in hood supports with regular boundaries. The only way to do that is to predefine excluded regions, which defeats the purpose of topology optimization by biasing creation of holes or pockets at particular locations. Manufacturing constraints are implemented as filters to remove designs that violate those constraints. Various Heaviside functions have been used as filters [36]. It is fair to say that today, it is not possible to get realistic manufacturable designs for mass-production processes directly from TopOpt.

2.2 Geometric Data Sets for Machine Learning

Training a machine learning algorithm requires datasets that account for various factors like quality, quantity, validity, and balance. These factors play an essential role in the overall performance of the algorithm. Large datasets in the fields of sales and business applications have been around for a while. However, in recent years, a few datasets have been generated for training ANNs (Artificial Neural Networks) in the engineering domain. The most common ones are the Modified NIST [10] and the Fashion-MNIST [35]. The MNIST and Fashion-MNIST datasets contain images of numbers and clothing apparel. Neither dataset contains data suitable for mechanical engineering or product design domains.

Methods have been developed to create datasets that include 3D data and corresponding annotations that include bounding boxes and viewpoints [11, 12, 15, 24]. These data sets are usually small (less than 1000 entities) or have fairly simple everyday objects that do not contain complex geometries, like the ones used in automotive body in white designs. Most datasets with viewpoint annotations are small in scale, coarse in viewpoint discretization and simple in scene context. The dataset developed by Xiang et al [34] has a large-scale database with 2D images and 3D shapes for 100 object categories. The objects have 3D annotations to objects by aligning the closest 3D shape to a 2D object. Koch et al [7] demonstrate a method to collect CAD data from a publicly available interface hosted by Onshape, post-process it, and convert it into representations suitable for machine learning algorithms. The size of the dataset is close to 1 million entities of everyday objects belonging to pre-defined categories [24]. The 3D CAD data obtained are in STEP format, which includes the BRep of the models. This data is further discretized to generate triangulated meshes that can then be fed into machine learning algorithms. However, the objects in the dataset are miscellaneous objects, not alternative designs to satisfy the same function and design specs.

In the engineering design domain, the use of data-driven design/automation is extremely desirable, requiring specific high-quality datasets which particularly difficult to come by. In such applications, data with higher resolutions, greater modality, and parameterization is required for learning useful relationships. Hence, engineering datasets inherently tend to be more customized to one specific application, requiring designers/researchers to curate the data required for each unique design space. Several such engineering datasets are the BIKED dataset [22] which includes assembly images, component images, design parameters, and labels for 4,500 unique bicycle designs. Chen et al. extended the UIUC Airfoil dataset [25] of nearly 1600 real airfoil designs to include aerodynamic lift and drag performance values [1]. Another dataset includes the microstructures for metamaterial systems curated by Wang et al. [31]. It also includes the associated tensor stiffness values. Wu et al. introduced another unique dataset called ModelNet [40]. The dataset was generated for a study of recognizing/generating a 3D shape based on a depth map image and contains over 150,000 3D voxel geometries across 660 unique categories. The CarHoods10k dataset [33, 21] has a more profound technical significance and applicability than some of the other datasets. CarHoods10k was generated and curated using methods outlined in Ramnath et al. [17, 16]. Specifically, it deals with automotive hood frame generation, but the workflow methods could be applied in virtually any engineering design domain.

Although these datasets are novel and valuable in their respective domains, they do not guide designers in creating a parameterized CAD model inspired by TO meshes. Hence a new dataset was curated that had a good balance between volume and variety, assisting designers with creating parameterized CAD models that mimic the performance of results obtained from TO. This dataset, along with the CarHoods10k dataset, enables training a machine learning algorithm effectively.

2.3 Machine Learning Algorithms

2.3 Machine Learning Algorithms The idea of developing machine learning algorithms is to enable machines to view and perceive features and patterns in training data and to use this knowledge for a multitude of tasks like image recognition, classification, natural language processing, etc. Advancements in deep learning has led to the development of Convolutional Neural Networks (CNNs). A CNN is a class of artificial neural network (ANN), most commonly applied to analyze visual imagery [29]. CNNs are based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation-equivariant responses known as feature maps [39].

Deep convolutional neural networks [8, 9] have led to a series of breakthroughs for image classification [9, 26, 38]. Deep networks naturally integrate low/mid/high-level features [38] and classifiers in an end-to-end multi-layer fashion, and the "levels" of features can be enriched by the number of stacked layers (depth). More recent studies show that network depth is critical, and the leading results [28, 3, 5] on the challenging ImageNet dataset [23] all require "very deep" models, with a depth of sixteen [28] to thirty [5].

Driven by the significance of depth, a question arises: Is learning better networks as easy as stacking more layers? An obstacle to answering this question was the notorious problem of vanishing/exploding gradients, which hamper convergence from the beginning [4]. Deeper networks suffer from a problem of degradation when they start to converge. An increase in depth, causes the accuracy to get saturated and then degrade rapidly [4]. Figure 3 compares the training and test errors of two networks (shallow vs deep) used on the same dataset.

The degradation problem is addressed by introducing a deep residual learning framework. Instead of hoping each few stacked layers directly fit a desired underlying mapping, these layers are allowed to explicitly fit a residual mapping [4]. In other words, the network can perform a 'skip connection' identity mapping as shown in Fig. 4. This identity mapping does not have any parameters and is just there to add the output from the previous layer to the layer ahead (by skipping the current layer). In most cases, the output from the previous layer and input to the next layer will not have the same dimension. Hence, the identity mapping is multiplied by a linear projection to expand the channels to match the residual. The Skip Connections between layers

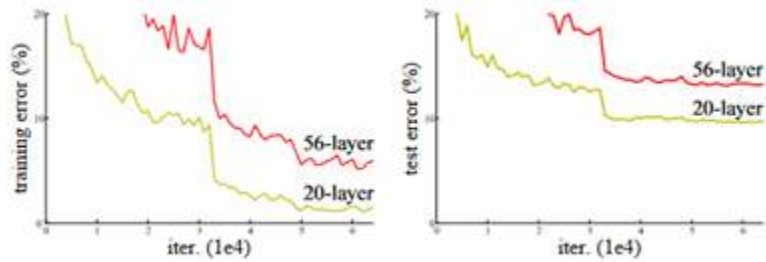


Figure 3: Training error (left) and test error (right) for 20-layer and 56-layer deep networks. The deeper network has higher training error, and thus test error [4]

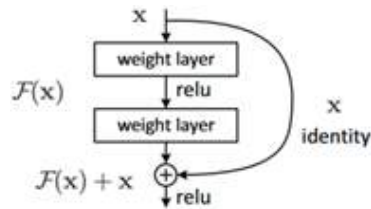


Figure 4: Residual learning: a building block [4]

add the outputs from previous layers to the outputs of stacked layers. This results in the ability to train much deeper networks than what was previously possible.

The architecture developed by He et al. [4] followed the following rules:

1. If the output feature maps have the same resolution e.g. $32 \times 32 \rightarrow 32 \times 32$, then the filter map depth remains the same
2. If the output feature map size is halved e.g. $32 \times 32 \rightarrow 16 \times 16$, then the filter map depth is doubled.

Given the drawbacks of plain or regular deep networks and the advantages of ResNet, the latter was used to develop an algorithm to perform image recognition and classification on the fuzzy dataset.

3 OVERVIEW

A high-level view of the data-centric system is shown in Fig. 5. It utilizes two large data sets of hood frames: one contains 10,000 parametric CAD models (Catia V5), and the other contains 6000 images of topology meshes of frames (fuzzy set). The term "fuzzy" here relates to the organic shapes obtained from TO. The contents and generation procedures are elaborated in later sections. It is envisioned that a hood frame designer is tasked with designing a support frame for a new hood skin. They will start by creating an offset solid of the skin to represent the available space for the frame, the upper boundary surface being the skin and the lower dictated by engine components. Topology optimization will then be conducted in some commercial software for one or more boundary conditions, such as hood lift, hood twist, frontal and side impact, and local impact. The resulting topology will be fed into a machine algorithm trained on the fuzzy data set. The ANN will try to find the closest matches of the topology to members of the fuzzy set. The seed design from the CAD data set corresponding to the matches found will be presented to the designer for review and selection. The selected design will be sent to another module that maps the feature pattern from the chosen CAD design to the new skin. This conformal mapping process is elaborated in a later section. Finally, an automatic parameterization

process will be used to produce a parametric CAD model, which the designer can analyze and modify as if it had been created from scratch.



Figure 5: Workflow of Data-centric Hood Frame Design

4 PARAMETRIC CAD DATA SET

A parametric CAD data set of 10,000 hood designs was generated in a previous project. The CarHoods 10k data set has been made available to the public [33, 21]. The details of procedures for generating this large set can be found in [17, 16, 32]. A quick overview is given in this section.

ML requires large data sets for training and validation. Apart from data volume, the set must be carefully curated to have variety and balance of relevant features/parameters. We started with six hood frames of actual hoods used in vehicles produced by our collaborators from the industry (Honda Design and Manufacturing Americas LLC). It was determined that 10,000 designs were needed for ML. Because many CAD models cannot be produced manually, the CAD model generation had to be automated. This necessitated the simplification of the geometry by omitting irrelevant features and idealizing curve and surface geometries. To confirm that the idealized geometries were still good surrogates for actual geometries, FE analyses of hood lift and twist under static loads were done on the six base models and their idealized counterparts. Several changes were made to the idealized CAD models to get them to yield deflection results that matched the base hood results within an acceptable range. The number of base models was increased to 10 in subsequent studies.

The next step was to enhance shape variety. This was achieved by mapping each of the 10 feature patterns of frames to each of the 10 skins, producing 100 shape variants (Fig. 6). The top row shows the 10 base skins. The diagonal in the 10x10 matrix represents the original frame on each base skin; the off-diagonals are feature patterns mapped from the original base to other skins. Originally, mapping was done with a combination of macros and user-defined features in a leading commercial CAD system [17]. This was specific to each type of feature pattern and required a constraint network to avoid feature-feature intersections. In this paper, we will present a generalized approach using Conformal Mapping based on interpolating functions.

The next step was to generate 100 size variants for each shape in Figure 7. This was done by varying feature size parameters using an optimal space-filling scheme for fractional factorial DOE with filtered parameters, giving 10,000 size and shape variants. Models were analyzed for key load cases using a standardized set of boundary conditions not affected by hood shape and size [13]. Since we used different CAD and FEA software, bridge software (CADNEXUS) was used to transfer parametric CAD models between the two. Thus, all 10,000 models in the dataset are associated with key FE results.

This dataset has also been used in conducting three non-traditional DOE-based optimization studies since the parametric sets vary with shape variants [19, 20, 27, 18].

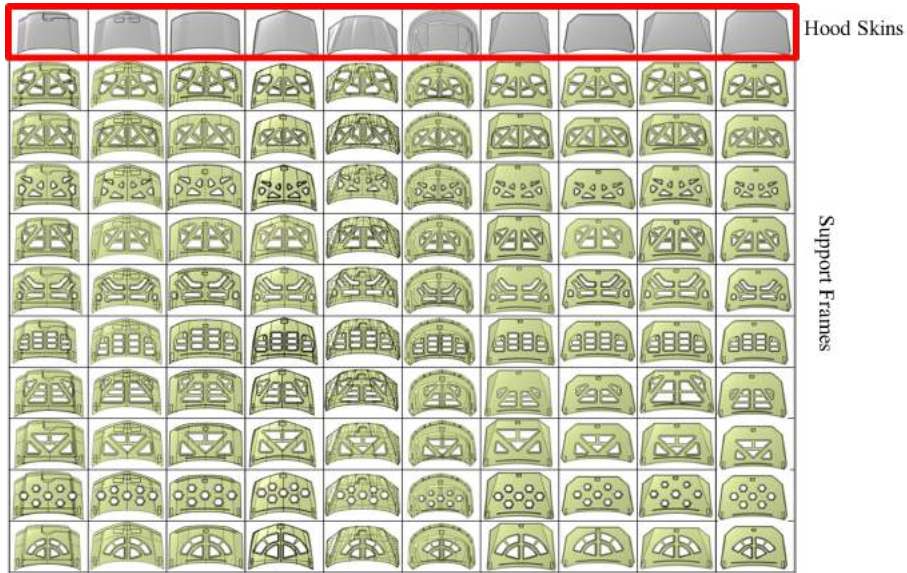


Figure 6: Shape variants of hood support frames [33]

5 FUZZY DATA SET



Another dataset is needed for matching TopOpt results to existing designs in the CarHoods10k set since the latter cannot be used directly for this purpose. CarHoods10k set contains precise BReps with regular boundaries of features, while TopOpt produces irregular and organic shapes in the form of meshes, even with disconnected regions and hanging "tails" (fuzzy features). The situation is analogous to ML for handwriting recognition where handwritten datasets (for example, EMNIST [2]) is used rather than typeset letters. The handwritten dataset contains squiggly lines as opposed perfect form in the typeset letters.

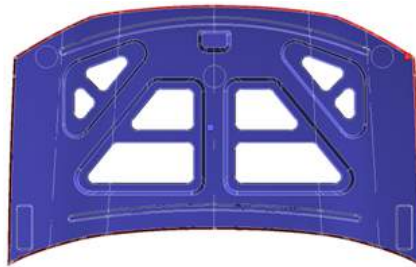
Any training data sets for machine learning must be adequate size, variety, and balance. The modality that the target machine-learning algorithm can handle must also be considered. The need for big data requires that automated procedures be implemented. The need for variety requires manipulating different topology optimization settings. The need for balance requires consideration of combinations of variants that span the optimization space well.

There were two approaches considered for generating the fuzzy feature data set. First, there was a skin-based domain system. To force the generation of pockets, it was necessary to use "starter holes", which essentially biases results. This would be a 'cheat' method of developing different topologies because changing the starter hole sizes and locations biases the final result. One could randomize hole locations to produce novel designs. However, there was a poor match between topology optimization results and existing hoods.

The other approach for generating the fuzzy feature database was using a support frame from the CAD dataset as the TopOpt domain region. The frame itself (a skin and feature pattern) was used in this approach. This approach used no starter holes. Boundary conditions and TopOpt parameters were manipulated to get a variety of resulting topologies. Further, adding or suppressing features of the frame randomly gave additional variety. Another advantage of this approach was that resulting topologies were already labeled with the seed from the CAD data set. Table 1 summarizes these observations. For these reasons, frame-based fuzzification was used to produce the fuzzy database.

Table 1: Comparison of Fuzzy Dataset Generation Methods

Skin based TO Domain	Frame based TO Domain
	
Produces novel designs	Limited by pockets and features
Poor match between TO results and existing hoods	Match between TO results and existing hoods is high
Requires additional clustering algorithms to assign labels	Ease of labeling "fuzzified" dataset based on skin and features

**Figure 7:** Hood Example (Outer Perimeter in Red)

5.1 Generating the Dataset - Geometries, Boundary Conditions, and Setup

The 100 shape variants shown in Fig. 6 were used as seeds for generating the fuzzy data set. Figure 7 shows feature types found on most frames: pockets in depressions, walls around pockets, front and back ribs, and support for hinges and locks.

FE boundary conditions for hood lift and twist were standardized to make them independent of skin size, shape, and curvatures. Three main support locations (hinge and lock support areas) were used. Hinges were simulated by having no remote displacement allowed except for Y-axis rotation. All DoFs of the lock were fixed. Standardized loads of 150N were applied to the front corners of the hood in the same direction to simulate lift and the opposite to simulate twist.

Sixty different topologies of each seed frame were generated by manipulating TopOpt settings and selective feature suppression, resulting in 6000 designs for the fuzzy database. The outer perimeter was specified as an exclusion region for TopOpt. Compliance minimization was used as the optimization objective. TopOpt setting manipulated were penalty factor (3-5) mass retention percentage (20-70%). Most hoods are symmetric about the longitudinal center plane. A manufacturing constraint of symmetry about the Y-Axis was added to account for this. A minimal member size of 20 mm was added for practicality to avoid tiny material slivers. Figure 8 shows a few fuzzy designs generated from CAD seeds.

Other methods, outside of Topology Optimization, were investigated. For instance, mesh morphing was considered but did not lead to feature variations. Another idea considered was direct editing, followed by meshing, of CAD models in the 10K dataset. This involved parametric variations of existing features and locations. It was difficult to automate for generating valid geometric models.

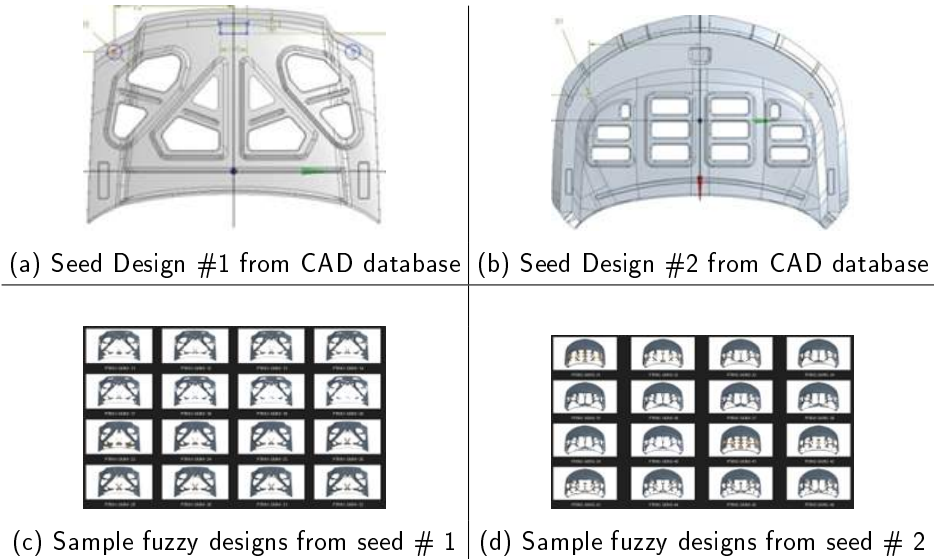


Figure 8: Subset of fuzzy designs generated

6 TRAINING ML ON FUZZY DATA SET

The decision to find suitable ML for training on the fuzzy set to match designer-produced topologies is coupled with the model modality to use. The results of TopOpt are 3D meshes, which can be expressed in various formats, such as STL. It is difficult to recognize 3D objects in their entirety; one approach used is to use orthographic projections to get 2D [6]. Another modality that could be used was pixelated images rather than geometric models. This latter modality is attractive because image recognition by ML is a well-developed field. From Fig. 8, one can see that top views of topologies capture a great deal of frame characteristics. Therefore, in this project, just the top view captured as a 2D image was used for training ML algorithms.

Screen captures were taken of the topologies and saved initially as high resolution (1473x758) *.png files. These images were cropped at outer boundaries to reduce image size. Some experimentation was done to find a lower resolution level without significantly losing features for enhanced efficiency.

6.1 ML Algorithm Selection

The capabilities of different Artificial Neural Networks were surveyed. A feedforward neural network (FNN) is unsuitable for image processing because too many neurons are needed, and spatial relations are lost. A convolutional neural network (CNN) assembles patterns of increasing complexity using smaller and simpler pattern encoded in their filters (kernels). Still, CNNs with many layers may suffer from vanishing gradient problems in backpropagation. Residual nets can skip layers in initial training, and then the network gradually restores the skipped layers as it learns the feature space, as described in section 2.3. Hence, the ResNet model was chosen over the plain deep network. During this project, two open-source ANNs were experimented with: ResNet(50) and ResNet(101) [37]. The 50 and 101 in the ResNet models refer to the number of convolutional layers in each model respectively. Further, serial and parallel machine learning networks were considered for feature recognition, but this was not pursued at this time; it may be of use in the future.

Similar to a regular CNN, ResNet also contains different filters that focus on different aspects while creating the feature map of an image. As the images progress through the layers, the details from the images slowly disappear, generating a pattern in the feature maps that human eyes cannot detect, but a neural network can.

The last set of fully connected layers are finally used to classify the input. Shown in Fig. 9 are various feature maps generated for a given input hood. Feature map 0 is generated at layer 1; feature map 10 is at layer 10 and so on. Each feature map enables the network to learn specific parameters or features in the input image.

The training was completed on the full dataset for both ResNet50 and ResNet101. The goal was to reduce the loss. The loss is used to compute the gradients, which are used to update the weights. Figure 10 compares the loss against the iteration number for ResNet50 and ResNet101.

As shown, at about iteration 10, ResNet50 performs just as well as ResNet101. Because of this, ResNet50 was chosen as the preferred network model for training the algorithm on the fuzzy feature dataset.

A limitation of the current state of the machine learning algorithm used in this research is that it only considers a top-level view. An option considered using multiple views in parallel, specifically, a section view of the hood, was proposed. Although this would aid in further improving the ML algorithm, the primary goal of this research is to develop a method for finding a preliminary match and then conformally map the features onto the new skin. Hence, the study of including additional views or developing a more complex ML algorithm has been left to future work.

7 CONFORMAL MAPPING

This section considers the following issue: once a topology-optimized result has been matched to a design in the CAD frame dataset, how are the features mapped onto a new skin? New skins will generally be of different sizes, aspect ratios and surface curvatures. For the features to be accurately mapped onto a new skin, it is important to account for the variations in the difference in aspect ratios and curvatures. This is done by using conformal mapping, which is a technique used in finite element analysis to map *perfect* elements onto irregular surfaces. A similar approach has been developed for mapping the features from an existing hood frame model onto a new skin. The following sub-sections will elaborate on conformal mapping and its use in hood frame design.

7.1 Mathematical Background

In mathematics, conformal mapping is defined as angle-preserving functions for curves transformed from one region to another. Conformal maps are function C that preserve the angles between the curves. More precisely: Suppose $f(z)$ is differentiable at z_0 and $\gamma(t)$ is a smooth curve through z_0 . To be concrete, let's suppose $\gamma(t_0) = z_0$ [14].

The function maps the point z_0 to $w_0 = f(z_0)$ and the curve γ to $\hat{\gamma} : \hat{\gamma}(t) = f(\gamma(t))$.

Under this map, the tangent vector $\gamma'(t_0)$ at z_0 is mapped to the tangent vector $\hat{\gamma}'(t_0) = (f \circ \gamma)'(t_0)$.

Angles are preserved, but distances are not. This concept is exploited in FEA for formulating stiffness equations based on perfectly shaped elements (squares, cubes, equilateral triangles) but using them for imperfect variations of topologically equivalent elements resulting from mesh generation of irregular bodies. For example, stiffness equations for a perfect square can be used for any QUAD element, with the caveat that accuracy varies with the extent of deviation. FEA implements a mapping from perfect to distorted shapes by establishing a local coordinate system independent of the actual size and shape. In 2D, the LCS (r, s) has its origin at the areal center; regardless of actual size, r and s vary between ± 1 . The corner nodes are assigned extreme values. For elements with mid-side nodes, one coordinate is taken as zero (Fig. 12).

FEA uses shape functions $N(r, s)$ to map interior local coordinates (r_i, s_i) to global coordinates (x, y) based on global nodal coordinates (x_i, y_i) as shown in Eq. 1, 2.

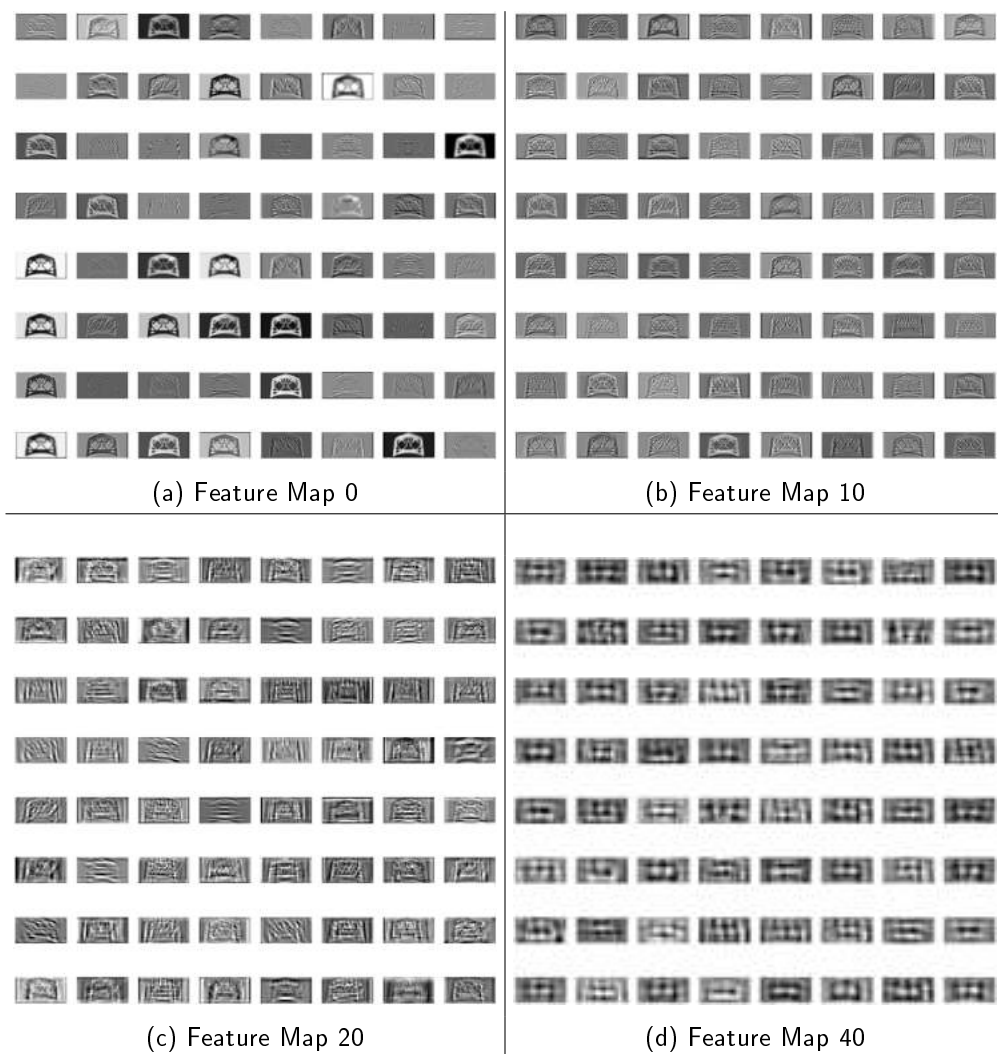


Figure 9: Example of Feature Maps for a specific hood at various layers

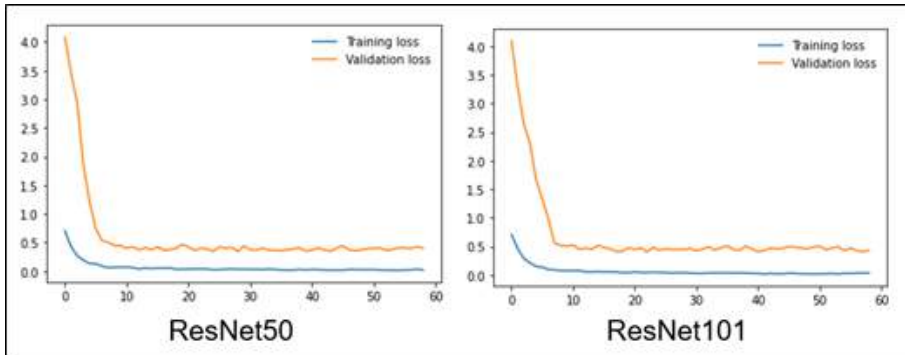


Figure 10: Loss vs Iteration Number, ResNet50 vs ResNet101

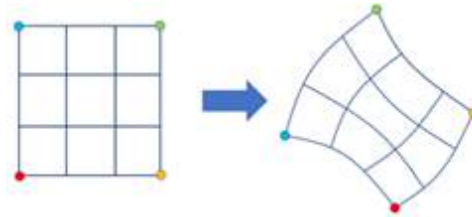


Figure 11: Conformal mapping example

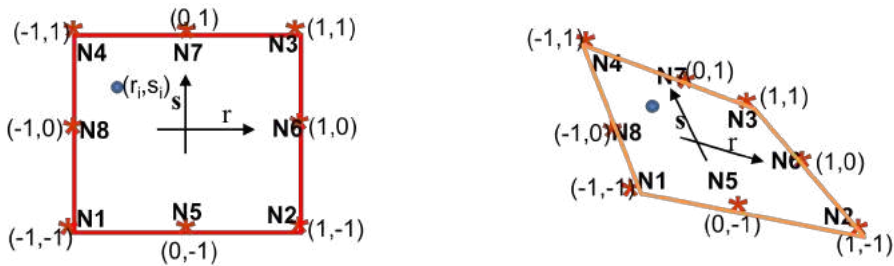


Figure 12: CS definition for QUAD element

Table 2: Shape functions for 8-noded QUAD element

$N1 = -0.25(1-r)(1-s)(r+s+1)$	$N5 = 0.5(1-r^2)(1-s)$
$N2 = +0.25(1+r)(1-s)(r-s-1)$	$N6 = 0.5(1-s^2)(1+r)$
$N3 = +0.25(1+r)(1+s)(r+s-1)$	$N7 = 0.5(1-r^2)(1+s)$
$N4 = -0.25(1-r)(1+s)(r-s+1)$	$N8 = 0.5(1-s^2)(1-r)$

$$x_i = \sum_j N_j(r_i, s_i)x_j \quad (1)$$

$$y_i = \sum_j N_j(r_i, s_i)y_j \quad (2)$$

where (r_i, s_i) and (x_i, y_i) are local coordinates of interior point i , and (x_j, y_j) are global coordinates of node j . Shape functions $N_j(r, s)$ correspond to node j and are evaluated at (r_i, s_i) .

7.2 Formulating Mapping Equations for Hood Features

The above mapping concepts were applied for mapping geometric features from one hood to another. It was observed that hood feature patterns have half symmetry. Therefore, only half the hood needs to be used in mapping, followed by mirroring the CAD model obtained about the symmetry plane. If the outer boundary of the hood is treated as if it were a Quad element, certain key points on it could be considered to be analogous to FE nodes. Key points on primary features, such as pockets, could be treated as interior points, and their positions relative to the nodes could be represented by local coordinates. These relative positions could then be mapped to a different hood using shape functions. However, in FEA, mapping only goes from local to global. For hoods, the local coordinates of key points must first be determined from the originating hood before mapping them to the target good. The latter will be referred to as "Forward Mapping", while the former is "Inverse Mapping". Forward mapping is identical to what is used in FEA, and those shape functions can be directly used. But Inverse Mapping equations need to be derived.

The symmetric half of hood skins shown in Fig. 13 do not have sharp corners or just 4 sides. However, one can manually identify boundary points that can be treated as corner nodes. Also, it is observed that some of the boundary edges are not straight. For this reason, it makes sense to use shape functions for higher-order Quads; mid-side nodes allow curves to be represented. The key points shown in Fig. 13, akin to FE nodes, will be referred to as Boundary Points (BPs).

There are as many shape functions as there are nodes. For 8-noded Quads, commonly used shape functions are listed below.

These functions can be used directly for Forward Mapping for finding global coordinates of point i using global BP coordinates by evaluating N_i at local (r, s) coordinates of any interior point i , as shown in Eq. 3

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} N1 & 0 & \dots & \dots & N7 & 0 & N8 & 0 \\ 0 & N1 & \dots & \dots & 0 & N7 & 0 & N8 \end{bmatrix} \begin{bmatrix} x1 \\ y1 \\ \dots \\ \dots \\ x8 \\ y8 \end{bmatrix} \quad (3)$$

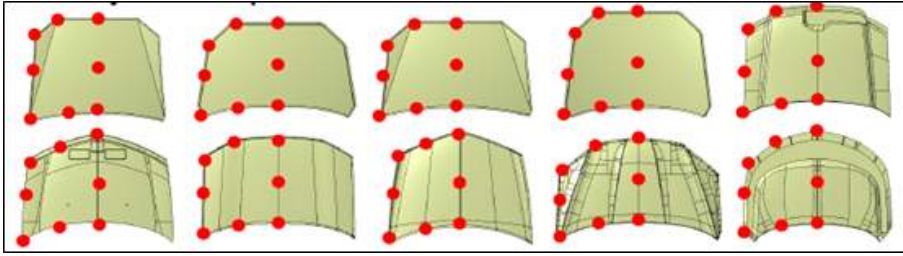


Figure 13: Boundary points of hoods (analogous to FE nodes)

In Inverse Mapping, global coordinates (x_i, y_i) of key feature points (KPs) are known in the originating hood. To find the local coordinates in that system, we need to write the above equations in terms of the unknown local coordinates (r, s) . This results in two quadratic equations in two unknowns r, s as shown in Eq. 4 and Eq. 5.

$$\begin{aligned}
 x_i = & -0.25 * (1 - r) * (1 - s) * (r + s + 1) * X_1 \\
 & + 0.25 * (1 + r) * (1 - s) * (r - s - 1) * X_2 \\
 & + 0.25 * (1 + r) * (1 + s) * (r + s - 1) * X_3 \\
 & + -0.25 * (1 - r) * (1 + s) * (r - s + 1) * X_4 \\
 & + 0.5 * (1 - r^2) * (1 - s) * X_5 \\
 & + 0.5 * (1 - s^2) * (1 + r) * X_6 \\
 & + 0.5 * (1 - r^2) + (1 + s) * X_7 \\
 & + 0.5 * (1 - s^2) * (1 - r) * X_8
 \end{aligned} \tag{4}$$

$$\begin{aligned}
 y_i = & -0.25 * (1 - r) * (1 - s) * (r + s + 1) * Y_1 \\
 & + 0.25 * (1 + r) * (1 - s) * (r - s - 1) * Y_2 \\
 & + 0.25 * (1 + r) * (1 + s) * (r + s - 1) * Y_3 \\
 & + -0.25 * (1 - r) * (1 + s) * (r - s + 1) * Y_4 \\
 & + 0.5 * (1 - r^2) * (1 - s) * Y_5 \\
 & + 0.5 * (1 - s^2) * (1 + r) * Y_6 \\
 & + 0.5 * (1 - r^2) + (1 + s) * Y_7 \\
 & + 0.5 * (1 - s^2) * (1 - r) * Y_8
 \end{aligned} \tag{5}$$

7.3 Mapping Implementation

The first step is to manually select BPs on baseline skins and KPs on baseline frame features. CAD models of 10 base hoods were used for this project. Eight BPs were defined on one-half of the base skins (Fig. 14). For some hoods, there is no clear corner point or mid-point, so one needs some judgment to predefine those on known skins. The midpoints were selected by using 0.5 geodesic length of side, as this would allow for a consistent selection on all hoods. In the study, 9 out of 10 skins were symmetric. The drawback of the

symmetry assumption is mitigated by the fact that the topology optimization used a y-symmetrical constraint, thus the data used in the training of the machine learning algorithm were symmetrical.

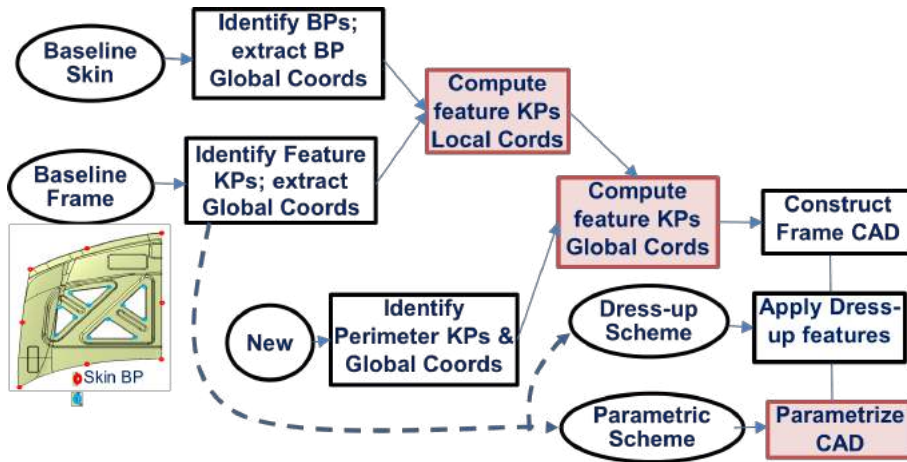


Figure 14: Mapping implementation

The next step was to select key frame features and define KPs for mapping. Each hood had unique features and patterns. Typical features were identified and classified. The distinction was made between primary and secondary (dress-up) features. Only the KPs of the primary need to be mapped, while dress-up features could be generated in CAD in relation to the primary. Dress-up parameters also need to be extracted (e.g., rib height).

The profile of features was approximated as polylines, straight and circular (Fig. 15). The former can be defined by 2 KPs, while the latter with 3 KPs (mid-points of each arc/corner and both tangent points). Using Inverse Mapping equations, the local coordinates of KPs are determined.

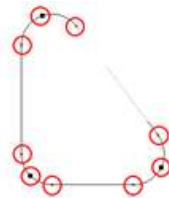


Figure 15: Feature boundary capture as a series of KPs

BPs need to be manually defined the first time a new target skin to which features are to be mapped, is considered. Automated procedures were implemented to Forward Map feature KPs to the new skin.

As seen in Fig. 16, the primary features are pockets and depression. CAD surface modeling with such features is created by sketching profiles and extrusion/cutout operations. Therefore, a generalized sketching macro (VBScript) was implemented that used KP global coordinates captured in Excel files and created feature profiles. Figure 17 shows an example of mapping profiles from a baseline skin-frame combination to a new skin. Macros designed to work within Catia V5 GSD workbench.

A second macro, the Builder macro, was implemented for sketch-based 3D operations: parallels, projections, translations, lofts, and fills. Projections, lofts, and fills are dimensionless; however, translations and parallels have dimensions. For these dimensions, the use of a scale factor was used while building the initially conformally





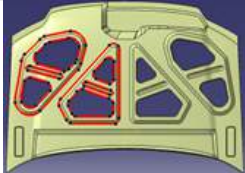



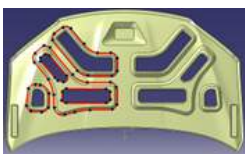
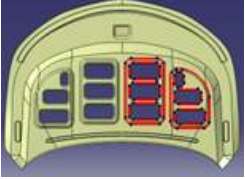
	Type: Triangles (3) & Parallelograms (4) #Pts: 50		Type: Triangles (3) & Parallelograms (1) #Pts: 26
	Type: Hexagons (4) #Pts: 24		Type: Triangles (1) & Parallelograms (3) #Pts: 30
	Type: Triangles (2) & Circular Fills (2) & Quadrilaterals (2) & Penta Laterals (2) #Pts: 56		Type: Triangles (6) & Circular Fills (2) & Middle Depression #Pts: 56
	Type: Triangles (4) #Pts: 24		Type: Triangles (5) & Circular Fills (2) & Depression #Pts: 58
	Type: Quadrilaterals (3) & Hex Lateral, Middle Pocket #Pts: 52		Type: Quadrilaterals (8) #Pts: 58

Figure 16: Key points for base frame primary features

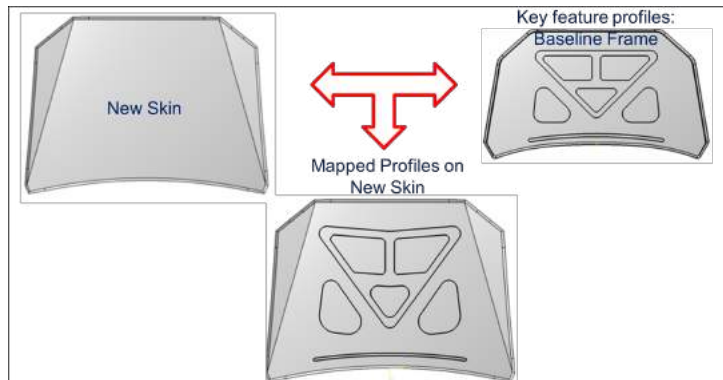
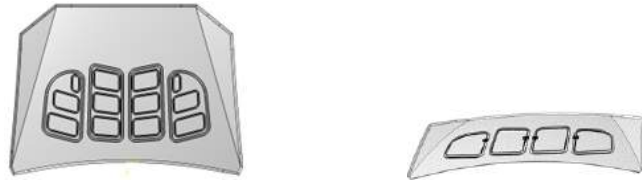


Figure 17: Mapping profiles



(a) Parallels and Projections Example (b) Projections and Translations

Figure 18: Subset of fuzzy designs generated

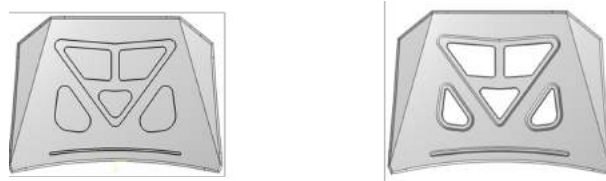


Figure 19: Builder Macro: from profile sketches to 3D features

mapped hood. The scale factor is the initial pattern's hood's skin area/new skin area. The cutouts in the surface model that represent the pocket features were defined by the innermost loop of the sketches used to define the ribs/walls.

The builder macro uses the sketches for the profiles to generate parallels (offset curves) needed for creating rib features. Then, the macro projects the required parallels and sketch profiles onto the new skin surface. Figure 18 shows examples of the operations performed by the builder macro. The depth/thickness of each rib is used to create offset curves from the surface. The parallel curves projected on the skin and the offset curves are lofted together to define the rib feature. The ribs created by lofting the curves resulted in open features and had to be closed using the fill surface operation. Figure 19 shows the builder macro result from key profile sketches.

8 MODEL PARAMETERIZATION

The ultimate objective of this project is to generate conceptual designs driven by topology optimization. Then designers can take the conceptual design and take it through detailed design. That requires modifying the geometry, adding and deleting features, and additional analyses. Therefore, it makes sense to produce a parametric CAD model that the designer can edit with key parameters. Editing is done at several levels:

- Local - pocket side dimensions, fillet dimensions, rib width and heights
- Global - This could move and rotate entire feature group or clusters
- Inter-cluster - This would allow changes of the relative distance between clusters
- Intra-cluster - This would allow for movements of pockets within a cluster.

Since feature parameters are extracted for use in the builder macro, they can be used for local editing. As shown in Fig. 20, there are 4 main parameters: pocket depth, rib depth, parallel 1 (inner rib), and parallel 2 (outer rib). As mentioned before, the initial auto-generation of the CAD model with the builder macro uses scaling of these parameters since they are not dependent on the 2D boundary points. This gave the ability of parametrization so that the user could adjust them after the hood was built.

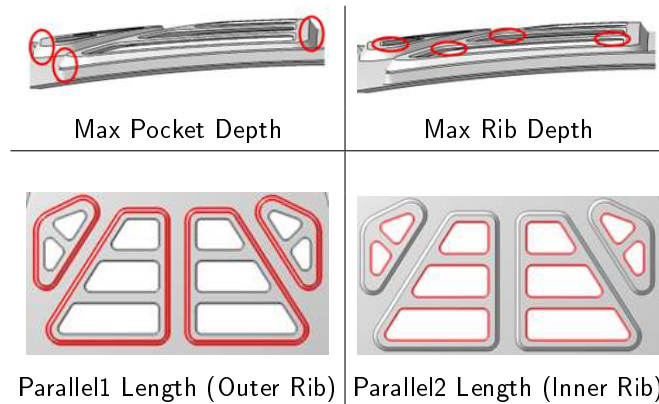


Figure 20: Edit-able local parameters

A cluster is a group of related features, such as multiple pockets in a depression with ribs of the same width and height around the perimeter (Fig. 21). Operations on clusters can facilitate the transformation of an entire group while preserving the dimensions and relative positions. This was done by pre-defining a 3D reference point (or cluster reference point) for all sketches. These 3D points are also projected with sketches. The cluster point can be used for global and inter-cluster positioning. Moving the cluster point will move all sketches that use it as a reference, but the sketches themselves have to ability to be shape-shifted based on designer requirements. There are two important assumptions: i) values for the profile position parameters are chosen such that they don't intersect each other, and ii) the profile sketches form closed loops.

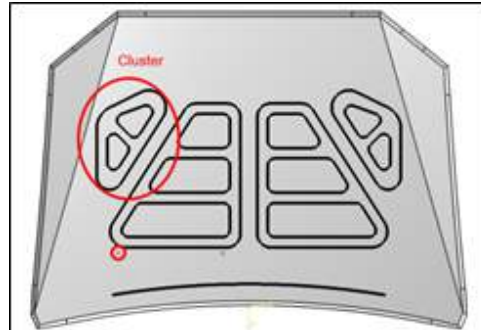


Figure 21: Feature cluster

9 CASE STUDIES

9.1 Application of Conformal Mapping to New Design

In the case of new skin (without the frame), the designer can perform topology optimization to obtain an analytical solution based on the boundary conditions and loads. In this case study, only the hood lift load case was considered since both datasets were created for this case. The new skin is not part of any of the current datasets. The first step is to obtain the results from TO and capture a top view of the same to be processed using the ML algorithm. Figure 22 shows the CAD model of the skin surface, the FE setup, and results obtained from topology optimization.

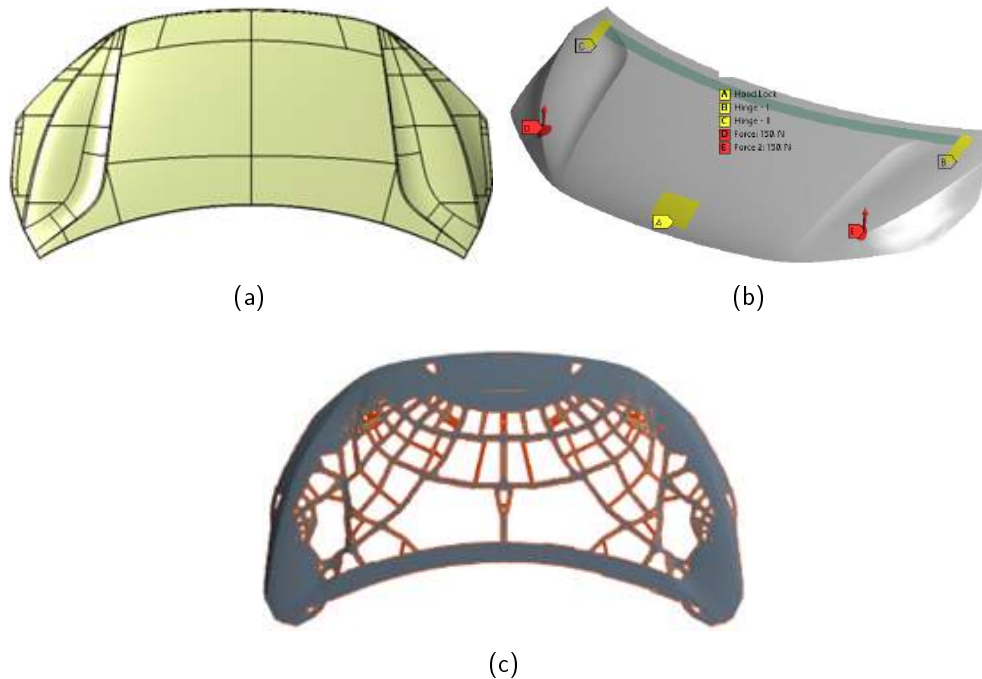


Figure 22: (a) Input new skin (b) FE setup for TO (c) Results from TO

The TO result is saved as a PNG file with a resolution of 1473 X 758 pixels. The result from topology optimization is saved as a top view image (Fig. 22(c)) which is then used to find the closest matches of that design in the fuzzy feature dataset. The ML algorithm trained based on the ResNet50 framework was used to predict the closest match. The top three predictions from the algorithm are summarized in Table 3. The last column ("Fuzzy Dataset") shows one sample from the fuzzy dataset for each combination of skin and pattern.

Table 3: Top three matches for the input skin in the fuzzy dataset

Rank	Percentage Match	Matched Skin# & PTRN#	Image of Match
I	68.18%	SKIN8 & PTRN3	
II	23.66%	SKIN10 & PTRN1	
III	2.3%	SKIN9 & PTRN3	

Based on the output from the ML algorithm, the most suitable feature pattern is PTRN3 which is used on

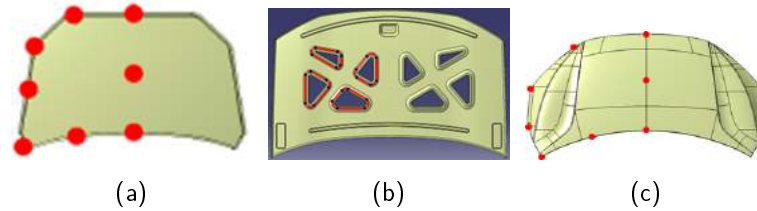


Figure 23: (a) Boundary points on SKIN8 (source) (b) feature points on PTRN3 (source) (c) boundary points on the new skin (target)

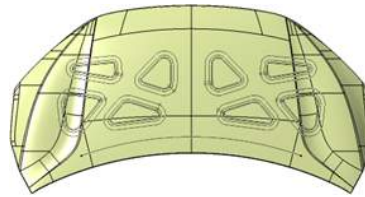


Figure 24: Profiles of PTRN3 mapped onto the new skin using macros

SKIN8. To conformally map PTRN3 from SKIN8 to the new skin, three sets of points have to be extracted. The first is BPs from SKIN8, second is KPs from PTRN3 and the third is BPs from the new skin. Figure 23 shows the BPs from SKIN8, KPs from PTRN3, and BPs from the new skin.

The profile of the PTRN features was approximated using polylines, and straight and circular arcs. As described in earlier sections, the local coordinates for the KPs are determined using the Inverse Mapping equations. The Forward Mapping equations then map the local coordinates of features in PTRN3 to global coordinates in the new skin using the corresponding BPs. The generalized sketching macro (VBScript) was then used to create the profiles of the features using the newly mapped KPs onto the new skin. Parameters were added during the creation of the sketches to enable the designer to make changes to feature profiles effortlessly. Figure 24 shows the profiles of PTRN3 created on the new skin.

The builder macro then performed its intended operations to create sketch-based 3D features: parallels, lofts, fills, pockets, etc. Parameters were also added during the generation of 3D features to control various aspects like the thickness of ribs, depth of ribs, etc. The parallel curves projected on the skin and the offset curves are lofted together to define the rib feature. The cutouts in the surface model that represent the pocket features were defined by the innermost loop of the sketches used to define the ribs/walls. The parameters can be updated to make changes to the profiles and/or position of the features. Figure 25 shows the builder macro result from key profile sketches and a few variations generated by modifying the parameter values.

9.2 Generalizability of the Approach

The class of design in aerospace and automotive structures using sheet metal stamped assemblies have common design features, even though the overall final design may be quite unique due to the combination, shape, and position of said features. The common features can then be grouped and classified using generic attributes making the design process more modular. The conformal mapping method allows designers to take advantage of the modular design process by using and combining features from one design into another.

The features on the door panel are represented using a combination of polylines, straight lines, and curved arcs. The specific positions (KPs) are established in local coordinates through the Inverse Mapping

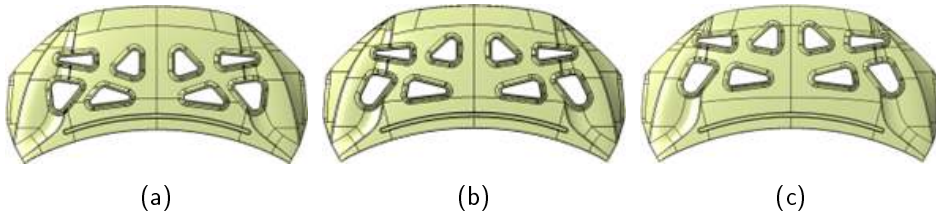


Figure 25: (a) Hood frame model after conformally mapping PTRN3 onto the new skin (b) modifying profile parameters (c) modifying position of features

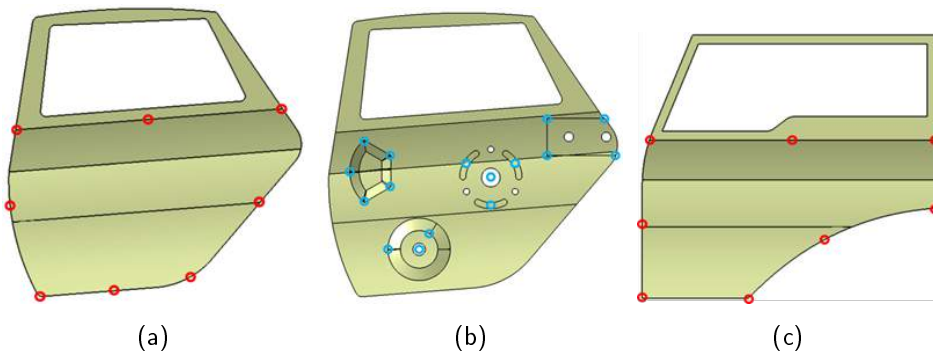


Figure 26: (a) Boundary points (BPs) on door panel I (source) (b) key points (KPs) on panel features (source) (c) boundary points (BPs) on the new door panel (target)

equations. Subsequently, the Forward Mapping equations translate the local coordinates of the source door panel's features to global coordinates on the new door panel surface using its corresponding BPs. To generate the feature profiles on the new surface, a sketching script (VBScript) was used, to incorporate the newly assigned KPs. Additionally, parameters were integrated into the sketch creation process, allowing the designer to make parametric updates (size, shape and position) to the feature profiles. Figure 27 shows the resulting profiles of features on the new surface.

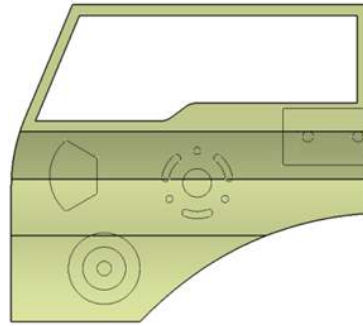


Figure 27: Profiles of features mapped onto the new door panel using macros

The builder macro generated the 3D features from the sketched profiles/outlines. These operations included creating parallels, lofts, fills, pockets, etc. Additionally, parameters were included in the generation of these 3D features to control features such as rib thickness and depth. The holes, representing cutouts in the surface model, were determined by the innermost loops of the sketches used to define the ribs or walls. By adjusting the parameters, it's possible to modify the profiles and/or positions of the features. Figure 28 illustrates the outcome of the builder macro, showing the results from key profile sketches along with some variations produced by tweaking the parameter values.

10 CONCLUSIONS

Prior to the development of Topology Optimization, the conceptual design of structural components was done in an ad hoc manner, based on intuition and experience. Topology Optimization represents an analytical

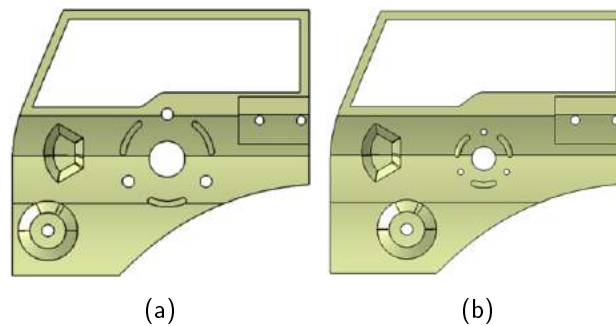


Figure 28: (a) Door panel model after conformally mapping the features onto the new panel (b) variation in feature size and position by updating parameters

approach that provides a guide on load paths where material needs to be reinforced and regions of low stress where material needs to be removed. However, the full potential of this mechanics-based approach is not realized because of limitations on manufacturability for mass production processes. This paper has demonstrated how analytical and experiential knowledge could be combined to produce better conceptual designs in the form of parametrized CAD models. These models are amenable to geometry modifications by designers and the addition of other features needed in detailed designs, such as specific geometries of hinges, locks, and Mastic features for adhesives. This approach relies on two large data sets, one of detailed CAD models generated from past designs and one of TopOpt meshes. The approach has been demonstrated in the context of a real-world complex structure, that of automotive hood support frames. But could be applied to other structural components, as well. The Conformal Mapping approach developed as part of this research can be used for transferring features and feature patterns from one component to another functionally equivalent one.

ACKNOWLEDGEMENTS

This research was funded by Honda Development & Manufacturing, Americas. The authors gratefully acknowledge financial and technical support for this work.

Satchit Ramnath, <http://orcid.org/0000-0002-2509-0626>

REFERENCES

- [1] Chen, W.; Chiu, K.; Fuge, M.D.: Aerodynamic design optimization and shape exploration using generative adversarial networks. In AIAA Scitech 2019 Forum, 2019. ISBN 9781624105784. <http://doi.org/10.2514/6.2019-2351>.
- [2] Cohen, G.; Afshar, S.; Tapson, J.; van Schaik, A.: EMNIST: an extension of MNIST to handwritten letters, 2017. <http://arxiv.org/abs/1702.05373>.
- [3] He, K.; Zhang, X.; Ren, S.; Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE International Conference on Computer Vision, 2015 Inter, 1026–1034, 2015. ISSN 15505499. <http://doi.org/10.1109/ICCV.2015.123>.
- [4] He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep residual learning for image recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2016-Decem, 770–778, 2016. ISBN 9781467388504. ISSN 10636919. <http://doi.org/10.1109/CVPR.2016.90>.
- [5] Ioffe, S.; Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. 32nd International Conference on Machine Learning, ICML 2015, 1, 448–456, 2015.
- [6] Kasaei, H.: OrthographicNet: A Deep Transfer Learning Approach for 3D Object Recognition in Open-Ended Domains, 2019. <http://arxiv.org/abs/1902.03057>.
- [7] Koch, S.; Matveev, A.; Jiang, Z.; Williams, F.; Artemov, A.; Burnaev, E.; Alexa, M.; Zorin, D.; Panozzo, D.: ABC: A Big CAD Model Dataset for Geometric Deep Learning. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 9593–9603. IEEE, 2019. ISBN 978-1-7281-3293-8. <http://doi.org/10.1109/CVPR.2019.00983>.
- [8] Krizhevsky, A.; Sutskever, I.; Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84–90, 2017. ISSN 15577317. <http://doi.org/10.1145/3065386>.
- [9] LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D.: Backpropagation Applied to Handwritten Zip Code Recognition. Neural Computation, 1(4), 541–551, 1989. ISSN 0899-7667. <http://doi.org/10.1162/neco.1989.1.4.541>.

- [10] LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.: Gradient-based learning applied to document recognition. In Proceedings of the IEEE, vol. 86, 2278–2323, 1998. ISSN 00189219. <http://doi.org/10.1109/5.726791>.
- [11] Leibe, B.; Schiele, B.: Analyzing appearance and contour based methods for object categorization. In 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings., vol. 2, 409–15. IEEE Comput. Soc, 2003. ISBN 0-7695-1900-8. <http://doi.org/10.1109/CVPR.2003.1211497>.
- [12] Lopez-Sastre, R.; Redondo-Cabrera, C.; Gil-Jimenez, P.; Maldonado-Bascon, S.: ICARO: Image Collection of Annotated Real-world Objects, 2010. <http://agamenon.tsc.uah.es/Personales/rlopez/data/icaro>.
- [13] Ma, J.: Comparative Study of Structural Optimization Methods for Automotive Hood Frames. Ph.D. thesis, The Ohio State University, 2020.
- [14] Orloff, J.: Complex variables with applications. In Geometric Definition of Conformal Mapping, chap. Complex va. MIT Open Courseware, 2018.
- [15] Ozuysal, M.; Lepetit, V.; Fua, P.: Pose estimation for category specific multiview object localization. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, 778–785. IEEE, Miami, FL, USA, 2009. ISBN 978-1-4244-3992-8. <http://doi.org/10.1109/CVPR.2009.5206633>.
- [16] Ramnath, S.; Haghghi, P.; Kim, J.H.; Detwiler, D.; Berry, M.; Shah, J.J.; Aulig, N.; Wollstadt, P.; Menzel, S.: Automatically Generating 60,000 CAD Variants for Big Data Applications. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2019.
- [17] Ramnath, S.; Haghghi, P.; Ma, J.; Shah, J.J.; Detwiler, D.: Design Science meets Data Science: Curating Large Design Datasets for Engineered Artifacts. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2020.
- [18] Ramnath, S.; Li, A.; Shah, J.J.; Detwiler, D.: Design exploration and prediction of automotive hood designs based on non-uniform feature parameters. In Proc. NAFEMS World Congress, 2021.
- [19] Ramnath, S.; Ma, J.; Shah, J.J.; Detwiler, D.: Intelligent Design Prediction Aided by Non-uniform Parametric Study and Machine Learning in Feature Based Product Development. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2021.
- [20] Ramnath, S.; Shah, J.J.; Detwiler, D.: A Comparative Design Study of Topologically Dissimilar Automotive Structures. Computer-Aided Design and Applications, 374–391, 2023. ISSN 16864360. <http://doi.org/10.14733/cadaps.2024.374-391>.
- [21] Ramnath, S.; Shah, J.J.; Wollstadt, P.; Bujny, M.; Menzel, S.; Detwiler, D.: OSU-Honda automobile hood dataset (CarHoods10k). In Dryad Dataset, 2022. <http://doi.org/10.5061/dryad.2fqz612pt>.
- [22] Regenwetter, L.; Curry, B.; Ahmed, F.: BIKED: A Dataset for Computational Bicycle Design With Machine Learning Benchmarks. Journal of Mechanical Design, Transactions of the ASME, 144(3), 2022. ISSN 10500472. <http://doi.org/10.1115/1.4052585>.
- [23] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A.C.; Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge, 2014. <http://arxiv.org/abs/1409.0575>.
- [24] Savarese, S.; Li Fei-Fei: 3D generic object categorization, localization and pose estimation. In 2007 IEEE 11th International Conference on Computer Vision, 1–8. IEEE, 2007. ISBN 978-1-4244-1630-1. <http://doi.org/10.1109/ICCV.2007.4408987>.
- [25] Selig., M.S.: UIUC Airfoil Data Site, 2012. http://www.ae.illinois.edu/m-selig/ads/coord_database.html#N.

- [26] Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. 2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings, 2014.
- [27] Shah, J.J.: Structural Optimization across Dissimilar Topologies for Automotive Body Design. In NAFEMS America conference, 2022.
- [28] Simonyan Karen; Zisserman Andrew: Very deep convolutional networks for large-scale image recognition. In 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 14, 2015. https://www.researchgate.net/publication/265385906_Very_Deep_Convolutional_Networks_for_Large-Scale_Image_Recognition%0Ahttp://www.robots.ox.ac.uk/.
- [29] Valueva, M.V.; Nagornov, N.N.; Lyakhov, P.A.; Valuev, G.V.; Chervyakov, N.I.: Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*, 177, 232–243, 2020. ISSN 03784754. <http://doi.org/10.1016/j.matcom.2020.04.031>.
- [30] Vatanabe, S.L.; Lippi, T.N.; Lima, C.R.; Paulino, G.H.: Topology Optimization with Manufacturing Constraints: A Unified Projection-Based Approach. *Advances in Engineering Software*, 100(1), 97–112, 2016.
- [31] Wang, L.; Chan, Y.C.; Ahmed, F.; Liu, Z.; Zhu, P.; Chen, W.: Deep generative modeling for mechanistic-based learning and design of metamaterial systems. *Computer Methods in Applied Mechanics and Engineering*, 372, 113377, 2020. ISSN 00457825. <http://doi.org/10.1016/j.cma.2020.113377>.
- [32] Witek, K.: Fuzzy Feature Recognition on Topology Optimized Meshes to Match Hood Parametric CAD models in a Curated Database to Aid in Conceptual Design. Ph.D. thesis, The Ohio State University, 2022.
- [33] Wollstadt, P.; Bujny, M.; Ramnath, S.; Shah, J.J.; Detwiler, D.; Menzel, S.: CarHoods10k: An Industry-grade Data Set for Representation Learning and Design Optimization in Engineering Applications. *IEEE Transactions on Evolutionary Computation*, 1–1, 2022. ISSN 1089-778X. <http://doi.org/10.1109/TEVC.2022.3147013>.
- [34] Xiang, Y.; Kim, W.; Chen, W.; Ji, J.; Choy, C.; Su, H.; Mottaghi, R.; Guibas, L.; Savarese, S.: ObjectNet3D: A Large Scale Database for 3D Object Recognition. In *European Conference on Computer Vision*, 160–176, 2016. http://doi.org/10.1007/978-3-319-46484-8_10.
- [35] Xiao, H.; Rasul, K.; Vollgraf, R.: Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, 2017. <http://arxiv.org/abs/1708.07747>.
- [36] Xu, S.; Cai, Y.; Cheng, G.: Volume preserving nonlinear density filter based on heaviside functions. *Structural and Multidisciplinary Optimization*, 41(4), 495–505, 2010. ISSN 1615-147X. <http://doi.org/10.1007/s00158-009-0452-7>.
- [37] Yamazaki, M.; Kasagi, A.; Tabuchi, A.; Honda, T.; Miwa, M.; Fukumoto, N.; Tabaru, T.; Ike, A.; Nakashima, K.: Yet Another Accelerated SGD: ResNet-50 Training on ImageNet in 74.7 seconds, 2019. <http://arxiv.org/abs/1903.12650>.
- [38] Zeiler, M.D.; Fergus, R.: Visualizing and understanding convolutional networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8689 LNCS(PART 1), 818–833, 2014. ISSN 16113349. http://doi.org/10.1007/978-3-319-10590-1_53.
- [39] Zhang, W.; Itoh, K.; Tanida, J.; Ichioka, Y.: Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Applied Optics*, 29(32), 4790, 1990. ISSN 0003-6935. <http://doi.org/10.1364/ao.29.004790>.
- [40] Zhirong Wu; Song, S.; Khosla, A.; Fisher Yu; Linguang Zhang; Xiaoou Tang; Xiao, J.: 3D ShapeNets:

A deep representation for volumetric shapes. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1912–1920. IEEE, 2015. ISBN 978-1-4673-6964-0. <http://doi.org/10.1109/CVPR.2015.7298801>.