Algorithm 1: Custom floating-point quantization

Input: $MODEL$ as the CNN
Input: E_{size} as the target exponent bit size
Input: M_{size} as the target mantissa bits size
Input: $STDM_{size}$ as the IEEE 754 mantissa bit size
Output: $MODEL$ as the quantized CNN

foreach *layer in* $MODEL$ **do**
  if *layer is Conv2D or SeparableConv2D* **then**
    $filter, bias \leftarrow GetWeights(layer)$
    foreach x *in* $filter$ *and* $bias$ **do**
      $sign \leftarrow GetSign(x)$
      $exp \leftarrow GetExponent(x)$
      $fullexp \leftarrow 2^{E_{size}-1} - 1$ `// Get full range value`
      $cman \leftarrow GetCustomMantissa(x, M_{size})$
      $leftman \leftarrow GetLeftoverMantissa(x, M_{size})$
      if $exp < -fullexp$ **then**
        $x \leftarrow 0$
      else
        if $exp > fullexp$ **then**
          $x \leftarrow (-1)^{sign} \cdot 2^{fullexp} \cdot (1 + (1 - 2^{-M_{size}}))$
        else
          if $2^{STDM_{size}-M_{size}-1} - 1 < leftman$ **then**
            $cman \leftarrow cman + 1$ `// Above halfway`
            if $2^{M_{size}} - 1 < cman$ **then**
              $cman \leftarrow 0$ `// Correct mantissa overflow`
              $exp \leftarrow exp + 1$
          $x \leftarrow (-1)^{sign} \cdot 2^{exp} \cdot (1 + cman \cdot 2^{-M_{size}})$
    $SetWeights(layer, filter, bias)$